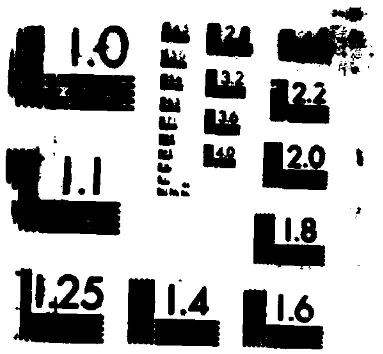
AD-A188 831   A DIGITAL CIRCUIT DESIGN ENVIRONMENT(U) AIR FORCE INST   1/2
OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING
C A ADAMS DEC 87 AFIT/GCS/ENG/87D-1

UNCLASSIFIED   F/G 9/1   NL



MICROCOPY RESOLUTION TEST CHART

A DIGITAL CIRCUIT DESIGN ENVIRONMENT

THESIS

Charles A. Adams Jr.
Captain, USAF

AFIT/GCS/ENG/87D-1

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

1

AFIT/GCS/ENG/87D-1

A DIGITAL CIRCUIT DESIGN ENVIRONMENT

THESIS

Charles A. Adams Jr.
Captain, USAF

AFIT/GCS/ENG/87D-1

Approved for public release; distribution unlimited

AFIT/GCS/ENG/87D-1

A DIGITAL CIRCUIT DESIGN ENVIRONMENT

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Computer Systems

Charles A. Adams Jr., B.A.
Captain, USAF

December 1987

Approved for public release; distribution unlimited

Preface

A need existed to simplify the interface between the students of the pregraduate digital design lab, and the Logic Simulator (LOGSIM) and Interconnect Expert (ICE) programs. These two programs provide valuable information on digital circuits; however, they did not provide the inexperienced user an easy approach. The solution was the development of this user friendly graphics oriented workstation which provides a circuit design environment. A computer listing of the graphics interface program is not included in this document; however, it can be obtained through the Air Force Institute of Technology, School of Engineering, Wright-Patterson AFB OH 45433.

The time that went into this thesis effort would have been overwhelming if not for my faculty advisor, Captain Bruce L. George. His help and reassurance made this work a success. A very special thanks is also due to Bruce Clay, whose knowledge, help, and graphics tool program made it possible to complete this thesis in the time allocated. Additional thanks goes to Captains W. H. Shaw and N. J. Davis for their assistance as members of my thesis committee. Finally, I wish to thank my wife Tammy for all her patience and personal sacrifice during all those long trying times, in the accomplishment of this thesis effort.

Charles A. Adams Jr.

Table of Contents

Page

List of Figures

List of Tables

Abstract

This thesis effort outlin~s the design and implementation of a graphics oriented digital design environment. This graphics environment provides the user with the tools to design digital circuits and then interface the user's circuit design with other tools (i.e., the Logic Simulator (LOGSIM) and Interconnect Expert (ICE) programs). This research paper presents the reasoning for the development of such a tool. The development of this tool involves reviewing database designs, specifically those oriented toward Computer Aided Design (CAD), and human-computer interface considerations. The paper also presents a recap of the features necessary for the graphics oriented environment. The detailed design of the program is presented along with the description of the file structures, which provide the underlying database. The limitations encountered and the results of the testing are addressed. In addition, a comprehensive user's manual is included for the operation of the graphics oriented interface. Finally, a listing of recommended follow-on efforts are presented with the conclusions of this thesis effort.

A DIGITAL CIRCUIT DESIGN ENVIRONMENT

I. Introduction

Background

A need existed to enhance the capabilities and the human-computer interface to the Computer Aided Design (CAD) tools used by the students in the EENG 450 Lab at the Air Force Institute of Technology. The Logic Simulator (LOGSIM) and the InterConnect Expert (ICE) are two systems currently used and being enhanced by other thesis efforts to assist in digital logic design (1) (2). The Logic Simulator is a digital design tool currently available on the Zenith Z-100 and DEC 11/780 computers. LOGSIM has a limited library of TTL Integrated Circuit (IC) chips which can be used for the design of a circuit. "LOGSIM is designed to accept a software representation of the circuit, check pin connections and test the logic operation of the circuit" (3). The operation of the system is accomplished by specifying the interconnections of the IC chips using a table format. This approach makes it difficult for the user to picture the actual circuit configuration.

The InterConnect Expert is also a digital design tool and is available on the VAX minicomputer. ICE is an Expert System which identifies to the user any possible wiring errors in the circuit design. These wiring errors do not include logic based errors. The ICE system also has its own limited library of usable TTL IC chips. This tool shares the limitation of not providing the user with a picture of the actual circuit layout.

Problem Statement

A graphics oriented/user friendly interface which provides a means to input design specifications and update the types and descriptions of circuit components will improve the productivity of users working with LOGSIM and ICE. There exists a relation between a digital design which can be implemented on either the LOGSIM or ICE and that which is physically constructed on a circuit breadboard. This relationship is easier to grasp with the use of a graphics oriented interface because it allows the user to see the circuit configuration. A single database common to both systems would provide a means to store the output from the LOGSIM (or ICE) in a configuration which can then be saved for later use, modified for further testing or submitted to the other digital circuit evaluator. In addition a shared database would provide a means to introduce new integrated circuit components to both systems.

Scope

This effort involved the development and testing of a graphics oriented user interface for the LOGSIM and the ICE software program systems for use in the EENG 450 Lab. The interface manages a CAD database which is common to each of these systems. The interface is implemented on one of the graphics-capable Zenith-248 workstations. The interface allows users to input portions of their circuit information graphically rather than in the form of a table of component instantiations and interconnections. Also, the graphics interface displays the output of the expert system and circuit simulation modules. A user's manual containing the procedure necessary to create a circuit graphically, test the circuit using ICE, simulate the circuit using LOGSIM, save or delete the circuit, and modify an existing circuit was developed.

Assumptions

It is assumed that the users of this graphics oriented interface have a basic knowledge of designing and wiring an electrical circuit. It is also assumed users know how to operate a Zenith-248 workstation. In addition, the user needs access to a TTL IC description/characteristics book.

General Approach

This effort was accomplished in four stages. The first stage included a literature search to determine an appropriate CAD database to implement. This is critical since the structure of the database must be acceptable to both the LOGSIM and ICE systems. This stage also included the requirements and preliminary design phases of software engineering. Several meetings with the individuals conducting thesis efforts on enhancing the LOGSIM and ICE were conducted. A review of the current approaches to the human-computer interface was accomplished to streamline the input requirements. Extensive study on the graphical layout of the ICs was conducted to determine an effective method of displaying the chips. In particular, the requirements to display all connecting lines between IC chips or a portion thereof was considered. In addition, the method of graphical input (mouse, keyboard, light pen, etc.) was also reviewed. The second stage involved the design and implementation of a software program, written in the C language, to convert the graphical circuit design into an input file in the database. The features of this interface also include taking a circuit which is stored as an input file and properly formatting it for use by the LOGSIM or ICE program, based on the user's specification. The results from running the file in the LOGSIM (or ICE) is stored as an output file. This output file is then available for user review/manipulation. A manipulated file is stored as a new input file. The third stage consisted of the testing and evaluation of the software program. The final stage entailed the development of a user's manual for this graphics oriented interface.

Sequence of Presentation

Chapter 2 provides a review of the literature relative to CAD database design, human-computer interface, and database management systems for the Zenith 248 workstation. Chapter 3 identifies the statement of requirements, justification of these requirements, and the preliminary system design. Chapter 4 provides the detailed design of the graphics oriented user interface. The implementation and problems areas encountered by the program are outlined in chapter 5. Chapter 6 provides the testing and results (validation). Chapter 7 contains the conclusion and specifies recommendations for additional features and further research. Finally, the user's manual for the system is included as an appendix to this thesis.

II. Literature Review

The emphasis of this phase of the thesis effort involved the research on databases and the human-computer interface. The database research involved the review of current databases that support Computer Aided Design and the minimum data information needed to accomplish the graphical interface tool environment. The goal of the human-computer interface research was to identify the means of providing an acceptable and easily usable input to the graphics interface package.

Database

The selection or design of a database management system is an important consideration in the development of any computer system which must process large amounts of data. There currently does not exist an ideal CAD database design which meets all the needs of the designers. This lack of an ideal CAD database design prompted numerous articles and books to be written on possible design approaches.

The reason the selection of a database management system is critical in the development of the graphics oriented user interface is that there exists a need to store and retrieve information applicable to three different programs. Specifically, the LOGSIM program developed by W. Deloria (1), the ICE program developed by S. M. Wagner (2), and the graphics interface. The requirement that these systems share a database on a Zenith Z-248 workstation introduces the necessity that the DBMS contain the minimum required capabilities in order to conserve memory space and improve the response time. Unnecessary features (e.g., simultaneous use of the database by multiple users) introduce the need for extra storage space. To ensure the DBMS contains at least the minimum features necessary, a review of the current literature on database designs was accomplished. The key considerations follow.

One technique of database management is a file-processing approach. A file-processing approach relies on developing programs which make queries, additions to, or deletions from files. Using a file-processing approach to data management provides a method of customizing the data storage and retrieval to the user's requirements; however, H. F. Korth and A. Silberschatz, in their book Database System Concepts, identify the major disadvantages of using a typical file-processing approach (4:2-3). The six major disadvantages they identify are 1) data redundancy and inconsistency, 2) difficulty in accessing data, 3) data isolation, 4) multiple users, 5) security problems, and 6) integrity problems.

Data redundancy occurs when the same information is stored in different locations of the same database. This can occur when different programmers develop procedures which insert data into the database. Data inconsistency is the condition where copies of the data stored in the file no longer agree. Data redundancy and inconsistency can be precluded by proper selection and development of the files and interface procedures. Data redundancy and inconsistency can also occur in any DBMS which is not properly constructed.

Difficulty in accessing data is the problem associated when attempting to generate a new request to the existing database. The file-processing approach does not lend itself to allowing for flexible database manipulation; however, the database and queries for the graphics oriented user interface will be fixed and therefore a changing query capability is not required.

Data isolation is a problem when the method of data storage is different between files. Developing new application procedures to manipulate the data when using multiple storage methods becomes more time consuming and difficult. This disadvantage can be dealt with if proper consideration is given to the data retrieval and storage procedures.

The multiple users disadvantage refers to systems which have the ability for multiple users to have access to the database simultaneously. This disadvantage is not applicable to the graphics oriented user interface since the database will be restricted to the single user of the Zenith Z-248 workstation. However, if the database was centrally located and several workstations had access to the database, this disadvantage would need to be considered.

The security problems disadvantage refers to the ability to restrict unauthorized users from gaining access to the database. This disadvantage is very important in some instances; however, it is not a consideration of this thesis effort. Users wanting to secure their circuit design need only remove the files which contain their circuit specific data from the workstation memory.

Integrity problems arise when constraints are established in the system. When changes occur to these constraints, appropriate changes to the program's code must be made. These changes may be compounded should the constraint be based on data items in several files.

The use of the file-processing approach may be appropriate depending on the user's needs regardless of the six major disadvantages identified (e.g., limited memory available). However, the contents of the data files remains the important consideration. Raymond A. Lorie, in his research report Issues in Database for Design Applications (5), identified some basic concepts for storing the data needed for design applications. In his research effort he used a relational database to accomplish the requirement of storing and retrieving engineering data. His approach involved the storing of data corresponding to the line segments which were necessary to reconstruct graphical displays. Once a basic graphic is constructed, more elaborate/detailed graphics can be made using some basic designs (e.g., a TTL chip could be represented using numerous rectangles). This approach to data storage and retrieval appeared to be the most promising to my research effort and is further elaborated on in Chapter III. In addition, Lorie developed a hierarchical structure among the tables of the relational database. This provides a means to take advantage of complex objects being composed of simpler objects.

Another approach to CAD database design was presented in the article "Modeling and Managing CAD Databases," by M. A. Ketabchi and V. Berzins.

> The deficiencies of current DBMSs in design applications and in the modeling and managing of refinements, alternatives, and versions of assemblies were recognized early in the history of design automation. Many engineering organizations and researchers have sought solutions for achieving effective management of design databases (6:94).

The authors suggested that the deficiencies in existing DBMSs can be solved using one of the following approaches; 1) develop a new DBMS with all the required features, 2) enhance current systems by adding missing features, 3) inclose the DBMS in a layer of software to overcome deficiencies, and 4) develop a special-purpose file manager that treats the DBMS as an application (6:94). Their article also addressed the necessity of developing an architecture which ensures the consistency of refinements, which they call versions. Because the users of the graphics oriented user interface may conduct revisions/alterations to existing data files, a review of Ketabchi and Berzins' solutions was accomplished.

The approach the authors used involved designing their overall database architecture to be based on three types of databases (i.e., private, project, and parts). Each of these databases have restrictions on the users ability to extract information directly. However, this approach does not appear to be applicable to this thesis effort because their decision to use three databases ultimately was to allow multiple users to introduce different versions of the same design.

Human-Computer Interface

The design of any computer system is not complete without considering the method of interfacing with the user. This is not a new issue but one that is frequently overlooked. A possible reason for this lack of consideration might be best described by a statement made by D.S. Woffinden in his master's thesis.

> One of the first things one notices in setting about to create a human-computer interface is the lack of any concrete, structured procedure or method to guide the design process. (7:15-16)

In his thesis report, D.S. Woffinden, conducted a comprehensive review of human-computer interface design principles. He concluded from his review that there exists twelve general design principles. These principles are 1) determine the purpose of the system, 2) know the user, 3) identify resources available, 4) consider human factors, 5) determine the interface language, 6) consider the environment of operation, 7) design for evolution, 8) optimize training,

9) accommodate levels of experience, 10) use selection vs. entry, 11) be consistent, and 12) anticipate errors. Although these principles are self explanatory, the reader is referred to D.S. Woffinden's thesis for a further description of each of these principles. Several of these design principles have already been incorporated in the development of the graphics oriented user interface (e.g., the interface language).

Henry Ledgard, in his article "Misconceptions in Human Factors", stated that "past research on human engineering has often been misdirected" (8:22). The author further identified the following eight misconceptions: 1) the primary purpose of human factors is to help novices, 2) ease of learning implies ease of use, 3) users should help design their own systems, 4) menus are easier to use than commands, 5) good human engineering centers on a few key design issues, 6) on large systems, users will be comfortable with partial understanding, 7) human engineering is for later stages of development, and 8) human factors are chiefly a matter of taste (8:22). Seven of his eight misconceptions listed appear to be reasonable; however, the third misconception does not. The author's suggestion that the user should not help design the system is questionable. A system is only as good as its use and if the system does not meet the users needs, it will not be used. If the user is not kept included in the design, wasted effort is a possible result.

Summary

The purpose of this chapter was to provide a review of the current work in the areas of databases for CAD and human/computer interface issues. It was not intended that this chapter present a complete discussion on all the work done in CAD Database management systems design, but only identify the key issues which were relevant. This review provided the ground work for the requirements analysis identified in the next chapter.

III. Requirements Analysis and System Design

The aim of this chapter is to identify the preliminary requirements and establish the underlying structure of the graphics oriented user interface program. All projects have some form of constraint(s) established prior to the start; therefore, this chapter starts with the recap of these initial constraints.

Existing Constraints

The graphics oriented user interface is designed to be installed on a Zenith Z-248 workstation. The complete package, which consists of the graphics interface, ICE, and LOGSIM must be eligible for widespread distribution. In particular, as much as possible, there should be no license requirements needed for any part of the package. This requirement has impact on the selection of a database management system.

CAD Database

The purpose of this section is to identify the basic structure of the database (i.e., identify the relations and their attributes) for use on the Zenith Z-248 workstation. The approach used in the development of the set of relations, necessary to store the graphical interface data, is based on the concepts provided in the research report written by Raymond A. Lorie (5).

Basic Structure. The data needed for this project can be categorized into five types based on their purpose. These categories are General Information, Graphical Display, Electrical Continuity, ICE Interface and LOGSIM Interface. The General Information category consists of those relations or files whose function is to store information about the IC chips (e.g., characteristics, description, and display pattern) and the operation of the program (i.e., Help menus). The Graphical Display category includes those relations necessary to store the information which the user provides in creating the digital design circuit display. The Electrical Continuity category contains the relations whose purpose is to store the interconnection between the components in the circuit. The ICE interface category consists of the information which is not related to designing the circuit, but required to evaluate the circuit using ICE and the corresponding results. Finally, the LOGSIM interface category contains the data required to simulate the circuit using LOGSIM and the corresponding results. A more comprehensive look at each of these categories of relations follows.

General Information Relations. Several relations have been selected to store the necessary information about an IC chip (e.g., size and description). In addition, the files containing instructional information for each phase of the circuit design process are required. The information needed by the graphics oriented user interface includes:

a. TTL package name (e.g., SN 7400).

b. Number of pins on the TTL package (e.g., 14).

c. Description of the chip's function (e.g., multiplexer).

Table I displays the attributes of the TTL BODY relation. This relation provides the primary description of each of the IC chips, which are available for use in either the LOGSIM or ICE programs. The number of pins is an important attribute because it will be a factor in determining the graphical display of the chip. The authorized values for the number of pins attribute are predetermined.

Table I

TTL BODY RELATION

TTL NAME	# OF PINS
7400	14
7420	14
7493	14
74175	16
74378	16

The actual screen display is based on the specific icon. TTLs are the only icon which varies in shape/size and this variation is based on the number of pins. However, knowledge

of the number of pins does not provide the user with the information about the functionality of the TTL. This information is stored in the relation shown in Table II.

Table II

TTL DESCRIPTION RELATION

TTL NAME	DESCRIPTION
7400	Quad 2 Input Positive NAND Gate
7420	Dual 4 Input Positive NAND Gate
7493	Single 4 Bit Binary Counter
74175	Quad D-Type Flip Flop
74378	Hex D-Type Flip Flop

The last type of relation in this category is the one responsible for storing the instructions/description of the program's operations. Four help files provide the required information. The first help file provides the instructions and description of the overall operation of the program. The second help file provides guidance on building circuits. The third file provides ICE interface unique information. Finally, the fourth file provides the LOGSIM interface unique instructions.

Graphical Display Relations. Two relations have been identified to maintain the information necessary to construct a digital design circuit display. The information required about the circuit includes:

a. Name of the circuit.

b. Unique ID for each icon in the circuit.

c. Type of TTL packages used.

d. Position of TTL package or input port in the display.

e. Paths of links from pin connections.

f. Color of the icon or link.

Each user must identify a unique name for their circuit. This allows the program to store the data in the same database. A function of the interface is to allow the users to construct their circuits in modules and later merge the entire network together. The unique circuit name allows for this capability. In addition to providing a unique circuit name, the user must identify what TTL IC chips he/she wants to use and where they are to be positioned. The position and orientation of the chip or port on the screen is determined by the coordinates of a reference point . Therefore, the coordinates for this reference point are also stored in this relation. Table III shows the attributes for the icon location relation. The domain for ICON-ID is an integer number. The TITLE values are the same as the domain of TTL Name in the previous relations or the names for the input ports.

Table III

ICON LOCATION RELATION

CKT	ICON-ID	TITLE	X-PT	Y-PT	COLOR
1	0001	7400	5	35	RED
1	0002	7402	25	27	YELLOW
1	0003	7400	45	23	RED
1	0004	7402	45	45	BLUE
2	0001	7474	4	30	GREEN
2	0004	7400	35	24	GRAY
3	0001	7408	5	35	ORANGE
3	0002	7400	25	27	WHITE

The PIN CONNECT PATHS relation shown in Table IV stores the information about the links between the pins or external interface ports (e.g., power, ground, clock, other circuits). The line segments which are stored in this relation provide a means to construct the path of the wires in the circuit. The circuit name (CKT) allows for future circuit merging as stated earlier. The X1, Y1 data corresponds to the coordinate of the line segment's starting point. The X2, Y2 data corresponds to the coordinate of the line segment's terminating point. The data presented in Table IV demonstrates a case where the user has specified two links in circuit #1 and one link in circuit #2. The first link in circuit #1 has two bends and therefore is stored as three line segments. The first three entries in the table correspond to the first link in circuit #1.

Table IV

PIN CONNECT PATHS RELATION

CKT	X1	Y1	X2	Y2
1	25	25	25	30
1	25	30	35	35
1	35	35	15	25
1	10	5	12	20
2	2	34	16	26

Electrical Continuity Relation. One relation is all that is necessary to store the information about the interconnections of the TTL pins and other electrical components (e.g., power, ground, or clock). The attributes of this relation must store the following information:

a. Name of the circuit.

b. Source of the input.

c. Destination of the output.

The LOGSIM and ICE systems require a listing of the connections between the pins/external interface ports. The user is directed to connect pins based on identifying an output/external interface port first then the input/external interface port to which it is connected. Table V shows the PIN CONNECTIONS relation which provides this requirement. The domain and purpose of the circuit name (CKT) is the same as above. The OUT-ID and OUT-PIN# are attributes which identify the output/external interface ports. The values for

OUT-ID are the Chip-IDs mentioned above prefixed by (P)ackage for TTL IC chips or a user specified number prefixed by (I)nput for external input devices (e.g., clock or other circuits). The values for OUT-PIN# are the appropriate pin number for the TTL chip or 00 for external input devices. The IN-ID and IN-PIN# are attributes which identify the input/external interface ports. The values for IN-ID are the Chip-IDs mentioned above prefixed by (P)ackage for TTL IC chips or a user specified number prefixed by (O)utput for external output devices (e.g., other circuits). The values for IN-PIN# are the appropriate pin number for the TTL chip or 00 for external output devices.

Table V

PIN CONNECTIONS RELATION

CKT	OUT-ID	OUT-PIN#	IN-ID	IN-PIN#
1	P0001	03	P0002	05
1	P0001	05	P0003	02
1	I0002	00	P0002	01
1	P0001	03	I0001	00
2	P0002	01	P0002	10

ICE Interface Relations. Two relations are necessary to store the output information from an ICE Execution. The attributes of these relations must store the following information:

a. Name of the circuit.

b. Pin locations where links are missing.

c. End points of questionable links.

The first relation, denoted ICE OUTPUT QUES RELATION, is shown in Table VI and consists of the data necessary to identify links which are questionable. These links were established by the user and are included in the PIN CONNECTIONS relation. The attributes include the pin connections on each end of the questionable link and the circuit name.

Table VI

ICE OUTPUT QUES RELATION

CKT	OUT-ID	OUT-PIN#	IN-ID	IN-PIN#
1	P0001	03	P0002	05
1	P0001	05	P0003	02

The second relation, denoted ICE OUTPUT MISS RELATION, is shown in Table VII and consists of the data necessary to identify links which are missing. These links were not established by the user and are not included in the PIN CONNECTIONS relation. The attributes include the pin connections where a link should be connected and the circuit name.

Table VII

ICE OUTPUT MISS RELATION

CKT	ICON-ID	PIN#
1	P0001	12
1	P0001	07

LOGSIM Interface Relations. Three relations are necessary to store the required input and output data for the LOGSIM interface. The attributes of these relations must store the following information:

a. Name of the circuit.

b. Input data stream for a each input port.

c. Pin locations where data is to be monitored.

d. Output data for each of the monitoring points.

The first relation contains the data needed to specify the monitoring points. The required attributes include the circuit name and a pin description.

Table VIII

LOGSIM INPUT RELATION

CKT	ICON-ID	PIN#
1	P0001	12
1	P0001	07

The second relation contains the data needed to specify the input data stream used for simulation. The required attributes include the circuit name, the input port description, and the data stream.

Table IX

LOGSIM MONITOR RELATION

CKT	PORT-ID	TITLE	DATA STREAM
1	0001	EN	010001110010
1	0002	AR	110011000100
1	0003	ET	100010010010
2	0002	TV	10110101
2	0001	XY	00011000

The final relation contains the output data provided after the execution of the LOGSIM program. The required attributes include the circuit name, the monitoring point description, and the output data stream.

Table X

LOGSIM OUTPUT RELATION

CKT	ICON-ID	PIN#	DATA STREAM
1	P0001	12	1101001001
1	P0001	07	0010101010

Human-Computer Interface Requirements

To this point only what the program requires has been addressed, but the means of communicating the user's needs to the system has not been identified. In the introduction, the need for a user friendly graphics interface was mentioned. This "user friendly" interface is critical because the likelihood of the system's use is related to the user's perception of the program's ease of use. Numerous articles have been written on the subject of the human-computer interface with emphasis made on keeping the user motivated and the system flexible to the user's needs. A help screen for interactive support provides supplemental instructions, should they be necessary. The "mouse" is the primary method of inputting the digital circuit design because it provides the user with the ease involved with "point and place." The user needs only point to the location where he wishes a TTL chip, an external interface port, or circuit connection and the desired feature is "placed" there. The decision to use a mouse instead of other input devices (e.g., light pen, tablet, or touch screen) was predicated on availability and cost. The optimum "point and place" method would seem to be the touch screen; however, the accuracy of making desired circuit connections is related to the surface area of the screen contacted by the finger/pointer. The keyboard is necessary for entering data about TTL types (e.g., 7400), filenames, and other required text.

Features

The technique for managing the data and determining what interface hardware was needed were only a part of the task involved in this thesis effort. The remaining task is to identify the necessary tools which are required by the circuit design environment. Interviews with possible user's resulted in the environment requiring the following features:

First, the ability to add (or delete) TTLs as determined by the user. In addition, a listing of all the possible TTLs, with their functional description, should be available to the user.

Second, the ability to add (or delete) an external input port. This port provides the user with a method of entering external data. In addition, the user should be able to label the port.

Third, the ability to add (or delete) links between components of the circuit.

Fourth, the ability to color code links and components. This provides a means of easily tracing links. In addition, component function grouping (i.e., components with the same function use the same color) enhances user understanding of circuit.

Fifth, the ability to save and retrieve circuit designs. The user specifies the location where the circuit is to be stored (e.g., on the hard disk or a floppy disk).

Sixth, the ability to delete the existing circuit.

Seventh, the ability to get help from the program. This feature aids previous users familiar with the program, but who have not used it recently. This feature is not intended as a substitute for the user's manual, but is used as a supplement.

Eighth, the ability to execute and view the results of either an ICE evaluation or a LOGSIM simulation. Furthermore, the user can get a printout of the result, if desired.

Finally, the user terminates the program, but only after the user confirms this decision. The confirmation precludes accidental circuit design loss.

Summary

This chapter described the constraints established for the digital design package. In addition, the type of information the CAD database must maintain in order to create, reconstruct, or pass data to/from ICE and LOGSIM was presented. The database involved the partitioning of the necessary relations into five category types, based on their purpose. These categories were General Information, Graphical Display, Electrical Continuity, ICE interface, and LOGSIM interface. The human-computer interface requirements for the graphics oriented user interface were also addressed. The "mouse" was selected as the primary interface for the graphics oriented user interface because of its "point and place" ability. This chapter also included the minimum features needed to provide the circuit design environment.

IV. Detailed Design

The design approach entailed a menu driven user interface. The user would be introduced to the program's options via the main menu. The main menu contained the basic operational tools which the user needed to manipulate (e.g., saving or deleting work in progress, retrieving and modifying existing designs, etc.) the circuit design data. The main menu also provided the gateway to three other menus; the "Circuit Design" menu, the "ICE" menu, and the "LOGSIM Menu." Design changes were implemented using the options available through the "Circuit Design" menu. In addition, the user was provided the option of conducting tests on the circuit design using the LOGSIM and ICE programs, via the corresponding menus. The transition between the menus and their visual display was provided by the Graphics Program, developed by Bruce Clay (9). A comprehensive review of the options available in each menu, including the method of their implementation and a description of the files follows:

Main Menu

The "Main Menu" was composed of eight options (see Figure 1). The first option, "Design Circuit," provided an access to the "Circuit Design" menu. The "Retrieve Circuit" option provided the user the capability to retrieve a circuit design which had already been designed and saved at an early time; however, the user could not retrieve a circuit if a circuit design was in progress. The "Delete Circuit" option provided the user the capability to delete the current design in progress. The "Save Circuit" option provided the user the capability to save the circuit design which was in progress. The "ICE" option provided an access to the "ICE Menu." The "LOGSIM" option provided the access to the "LOGSIM Menu." The "Help" option provided the user with a recap of the function of each option in the "Main Menu." The final option, "Exit," provided the user with a method to terminate the program.

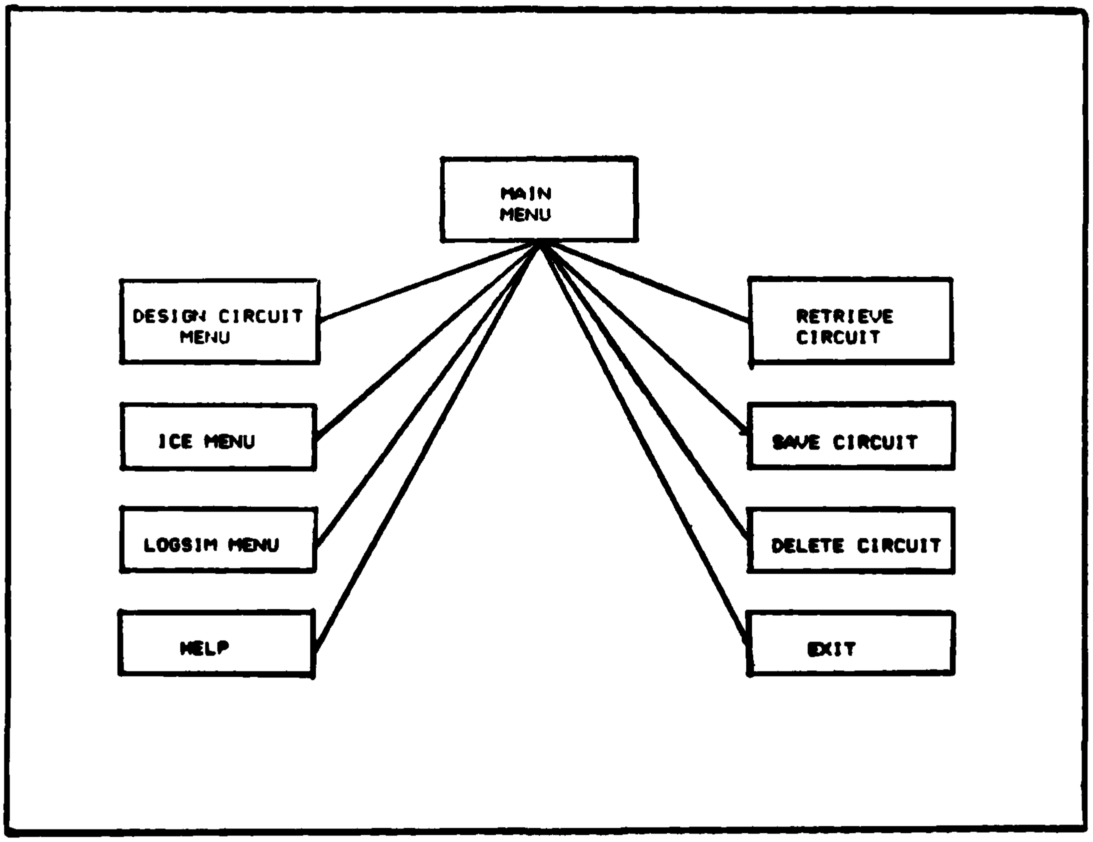


Figure 1. Main Menu Options

Design Circuit. The selection of this option resulted in the "CKT_IN_PROG" flag being set, if not already set, indicating a circuit design in progress. If the flag was set by this option, the initial circuit preparation was accomplished. Circuit preparation included the creation of the empty temporary circuit design files (i.e., files designated with the word "temp" having extensions ".loc," ".icn," ".grf," ".ckt," ".in," ".ind," ".dis," ".out," ".wav," ".iot," and ".txt"). The flag was used to preclude previously saved circuit design retrieval. In addition, the "Main Menu" options would then be replaced by the "Circuit Design" menu options (i.e., "Add TTL," "Del TTL," "Add Link," "Del Link," "Add Port," "Del Port," "Set Color," "Help" and "Main Menu") described below.

Retrieve Circuit. If this option was selected, a test was made to see if the "CKT_IN_PROG" flag was set. If the flag was not set, the user was then prompted to supply the filename of the previously saved circuit design. A test for the file was conducted and, if it existed, the files were copied into the temporary circuit design files, identified above. If the files did not exist or if the "CKT_IN_PROG" flag was set the user was provided the appropriate error message.

Del CKT. If this option was selected, the user was prompted for a confirmation. If the user confirmed the request (i.e., entered a "y" or "Y"), the circuit design window was cleared and the "CKT_IN_PROG" flag was no longer set. If the user did not confirm the request (i.e., entered any other input), no action was taken.

Save CKT. If the user selected this option, a test for the "CKT_IN_PROG" flag was made. If the flag was set, the user was prompted for a filename. If the user provided an invalid filename (e.g., more than eight characters or blank spaces), an error message was presented. Otherwise, a test was initiated to see if the file already exists. If the file existed, the user was prompted as to whether to overwrite the existing file. If the user specified not to overwrite the existing file, a request for a new filename was given and the process is repeated. If the file did not already exist or if the user wished to overwrite the existing file, a copy of the temporary circuit design files were made to the filename specified. If the "CKT_IN_PROG" flag was not set the user was provided an appropriate error message.

ICE. The selection of this option resulted in a test for the "CKT_IN_PROG" flag. If the flag was not set, an error message was displayed and the user remained in the main menu. If the flag was set, a test for the appropriate executable program was conducted. If the program did not exist, an error message was displayed and once again the user remained in the main menu. However, if the program was available and a circuit was in progress, the "Main Menu" options were replaced with the "ICE Menu" options (i.e., "Execute," "View Rslt," "Print," "Help," and "Main Menu") described below.

LOGSIM. The selection of this option resulted in a test for the "CKT_IN_PROG" flag. If the flag was not set, an error message was displayed and the user remained in the main menu. If the flag was set, a test for the appropriate executable program was conducted. If the program did not exist, an error message was displayed and once again the user remained in the main menu. However, if the program was available and a circuit was in progress, the "Main Menu" options were replaced with the "LOGSIM Menu" options (i.e., "Execute," "View Rslt," "Add Inpts," "Monitor," "Print," "Help," and "Main Menu") described below.

Help. The selection of this option resulted in the current screen display being replaced by a screen display containing a brief description of each of the "Main Menu" options. The user was prompted to press any key to continue. Once any key was depressed the "Help" screen display was replaced by the circuit design window.

Exit. The selection of this option resulted in the user being prompted for a confirmation of the request. If the user confirmed the request (i.e., entering a "Y" or "y"), the temporary files mentioned above were deleted and the program terminated. Otherwise, the selection of this option was ignored.

Circuit Design Menu

The "Circuit Design menu" was composed of nine options (see Figure 2).

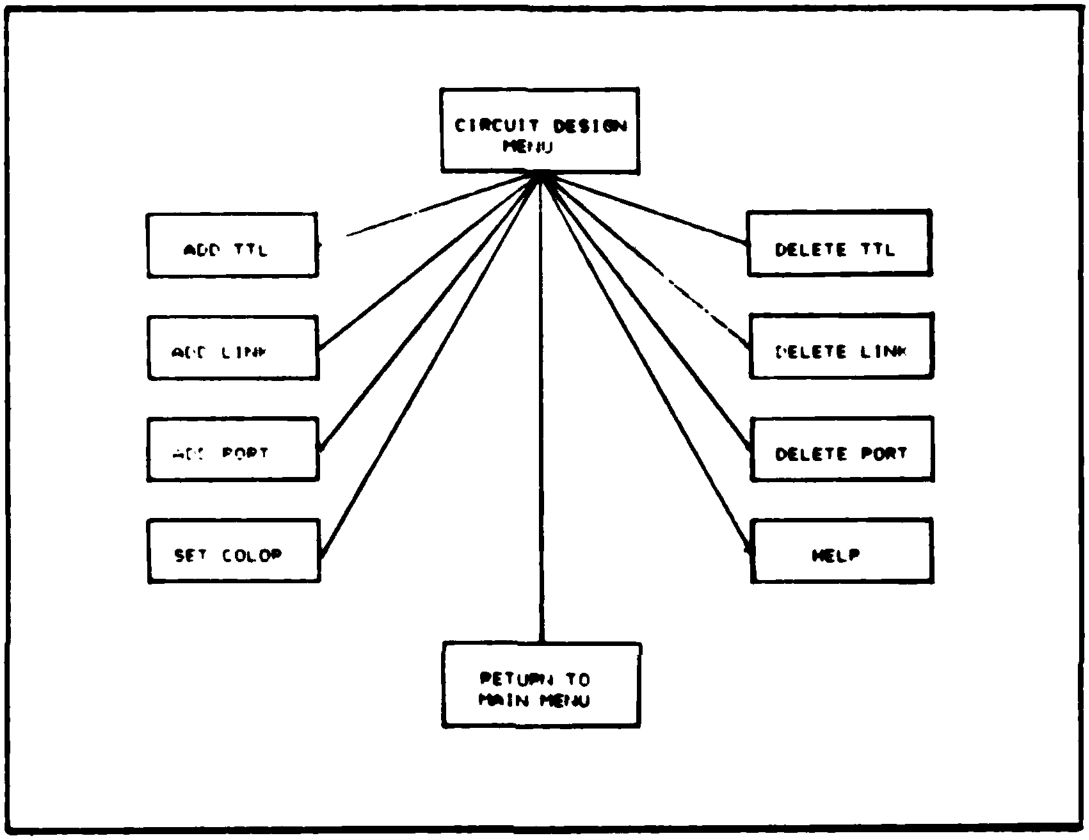


Figure 2. Circuit Design Menu Options

The first option, "Add TTL," provided a means of adding an integrated circuit (referred to in the graphics-oriented interface as a TTL) chip to the current circuit design. The "Del TTL" option provided a means of removing a TTL chip. The "Add Link" option provided a means of adding links between pin connections. The "Del Link" option provided a method of removing existing links. The "Add Port" option provide a means of adding external input ports to the circuit. The "Del Port" option provided a method to remove the input ports. The "Set Color" option allowed the user to change the color of subsequent circuit components. The "Help" option provided the user with a recap of the function of each option in the "Circuit Design" menu. The final option, "Main Menu," provided the user a method to return to the previous menu.

Add TTL. The selection of this option resulted in the user being prompted for a TTL type or a "?." The "?" entry resulted in the current screen display being replaced by a display of the thirty-two available TTLs and their description. This information was extracted from the "TTL.DES" file. To return to the circuit design screen the user pressed any key. The user was once again prompted for a TTL type. The user specified TTL type was compared against a list of the available TTLs and their pin count (stored in the TTL.DAT file). If the TTL was not in the file, a pin count of "0" was specified and the user was provided an error message. If the TTL was valid the user was advised to point to the location where the TTL was desired. Once the user selected the TTL placement, a test was conducted to see if the size of the TTL specified would overlap other components (e.g., TTLs, ports, power icon, ground icon, or clock icon) or the design window perimeter. If the placement resulted in an overlap, the user was given an error message and no TTL was placed. If the placement of the TTL did not cause an overlap, the TTL was placed in the design window. The TTL was constructed using a set of rectangles of predetermined size. In addition, the type and identifier for the TTL was annotated on the chip. The location of the TTL was then stored in the "temp.loc" file along with the dimension of reserve space. Each TTL was provided extra spacing on its right, left, and bottom sides for future wiring (link) paths. Furthermore, the type of icon (i.e., TTL), TTL id, TTL type (e.g., 7400), color, and reference coordinates for the TTL were stored in the "temp.icn" file. The "temp.loc" file is used when testing for possible overlap.

Del TTL. The selection of this option resulted in the user being prompted to point to any pin on the TTL that was to have been deleted. A test was then conducted to see if the pin was that of a TTL. If the user pointed to something other than the pin of a TTL, an appropriate error message was posted. This test was accomplished by having reviewed the coordinates of the current mouse position against the designated TTL locations, stored in the "temp.loc" file. If the pin belonged to a TTL, the next test was to see if the TTL had any links connected to it. This was accomplished by having reviewed the "temp.ckt" file (which contained the data on the links in the circuit). If the TTL had any connections, an appropriate error message was posted. If no links exist, the data on this TTL is removed from the "temp.loc" and "temp.icn" files. The screen was then cleared and the circuit was redrawn. Since no reference to the deleted TTL was in the files, the TTL was not redrawn.

Add Link. Links could only be started or ended at existing pin locations. The decision to start or terminate links at existing pin locations and not on links was based on the ability to identify a unique termination point. Pin locations are unique because icons are restricted from overlapping; however, links are not unique because they can overlap and yet not be connected, resulting in possible confusion. In addition, only the bends and terminating points of the links are stored, making it difficult to identify intermediate points on the line. If the user selected this option, a prompt was given to identify the starting pin location. Once the user clicked the left mouse button at the starting point, a test was conducted to see if the starting point was a pin location. This was accomplished by a review of all the pin locations (which is stored in the "temp.loc" file). If the user attempted to start or end at something other than a pin location, an error message was posted. If the user specified a valid starting point, the user was then prompted to identify the path of the link, by clicking the left mouse button at the turning points. Each subsequent left mouse button click identified a turning point and the line segment from the previous point was drawn. The drawing of the lines and points were a function of Bruce

Clay's graphics program (9). If the user attempted to establish a turning point outside the design window, an error message was posted and the user was requested to try again. If the user attempted to make more than ten turning points the line segments were not created. Limiting the number of bends to ten was a design/programming decision. To terminate the link the user had to click the right mouse button on a valid pin location. Once the path was established, two of the temporary circuit files (i.e., "temp.grf" and "temp.ckt") were updated. The link path, identification number, and color were stored in the "temp.grf" file. The "temp.ckt" file was updated to include the established connections.

Del Link. The user must provide the ending points of any link to be deleted. The reason for requiring the user to identify the terminating points versus any point on the link is directly related to the uniqueness mentioned above. The user is first prompted to click the left mouse button on either end of the desired link. A test is accomplished to confirm the location is a valid pin location (using the method described earlier). If the test fails, an error message is posted and the process terminated. If the location was a pin, the user is prompted to click the left mouse button on the pin at the other end of the link. If the user specifies something other than a valid pin location, an error message is posted and the process terminated. However, if both end points are valid pin locations, a test is conducted to determine if a link exists between these pins. This test is accomplished by reviewing the "temp.ckt" file (which contains the listing of all connections). If the connection exists, the link identification number is extracted form the "temp.ckt" file. The link is then removed from both the "temp.ckt" and "temp.grf" files using the link identification number. Once the files were updated, the design window was cleared. The circuit was then redrawn. Since no reference to the deleted link existed in the updated files, the link was not redrawn.

Add Port. The selection of this option resulted in the user being prompted to specify the port location. Once the user selected the port placement, a test was conducted to see if the specified port location would cause overlap of other components (e.g., TTLs, ports, power icon, ground icon, or clock icon) or the design window perimeter. If the placement resulted in an overlap, the user was given an error message and the option terminated. If the placement of the port did not cause an overlap, the user was prompted for a two-character name for the port. The reason for limiting the port label to two characters was because that's the maximum number of characters that could be displayed within the port icon. Once the user specified the two-character name, the port is displayed at the selected location using a set of rectangles of predetermined size. In addition, the two-character name was annotated on the port. The location of the port was then stored in the "temp.loc" file along with the dimension of reserve space. Each port was also provided extra spacing on its right, left, and bottom sides for future wiring (link) paths. Furthermore, the type of icon (i.e., port), port id, port name (e.g., I1), color, and reference coordinates for the port were stored in the "temp.icn" file.

Del Port. The user was prompted to point to the pin on the port that was to have been deleted. A test was then conducted to see if the specified pin belonged to a port. If the user pointed to something other than the pin of a port, an appropriate error message was posted. This test was accomplished by having reviewed the coordinates of the current mouse position against the designated port locations, stored in the "temp.loc" file. If the pin belonged to a port, the next test was to see if the port had any links connected to it. This was accomplished by having reviewed the "temp.ckt" file (which contained the data on the links in the circuit). If the port had any connections, an appropriate error message was posted. If no links exist, the data on this port was removed from the "temp.loc" and "temp.icn" files. The screen was then cleared and the circuit was redrawn. Since no reference to the deleted port was in the files, the port was not redrawn.

Set Color. The selection of this option resulted in the user being requested to point to the color on the screen which was desired for subsequent components. Once the user specified a location on the screen, a test was initiated to see if the color was the same as the background. If the color was the same as the background color, no change in color was made. However, if the user pointed to any other color on the screen, the current foreground color was changed to the selected color. This feature was provided in Bruce Clay's graphics program (9).

Help. The selection of this option resulted in the current screen display being replaced by a screen display containing a brief description of each of the "Circuit Design" menu options. The user was prompted to press any key to continue. Once any key was depressed the "Help" screen display was replaced by the circuit design window.

Main Menu. The selection of this option results in the "Circuit Design" menu options being replaced by the "Main Menu" options.

ICE Menu

The "ICE Menu" was composed of five options (see Figure 3). The first option, "Execute," provided the ability to execute the ICE program. The "View Rslt" option provided the user the method of seeing the results of the ICE execution. The "Print" option provided the user a means of getting a printout of the ICE results. The "Help" option provided the user a recap of the function of each of the ICE Menu options. The final option, "Main Menu," provided a means of returning to the previous menu.

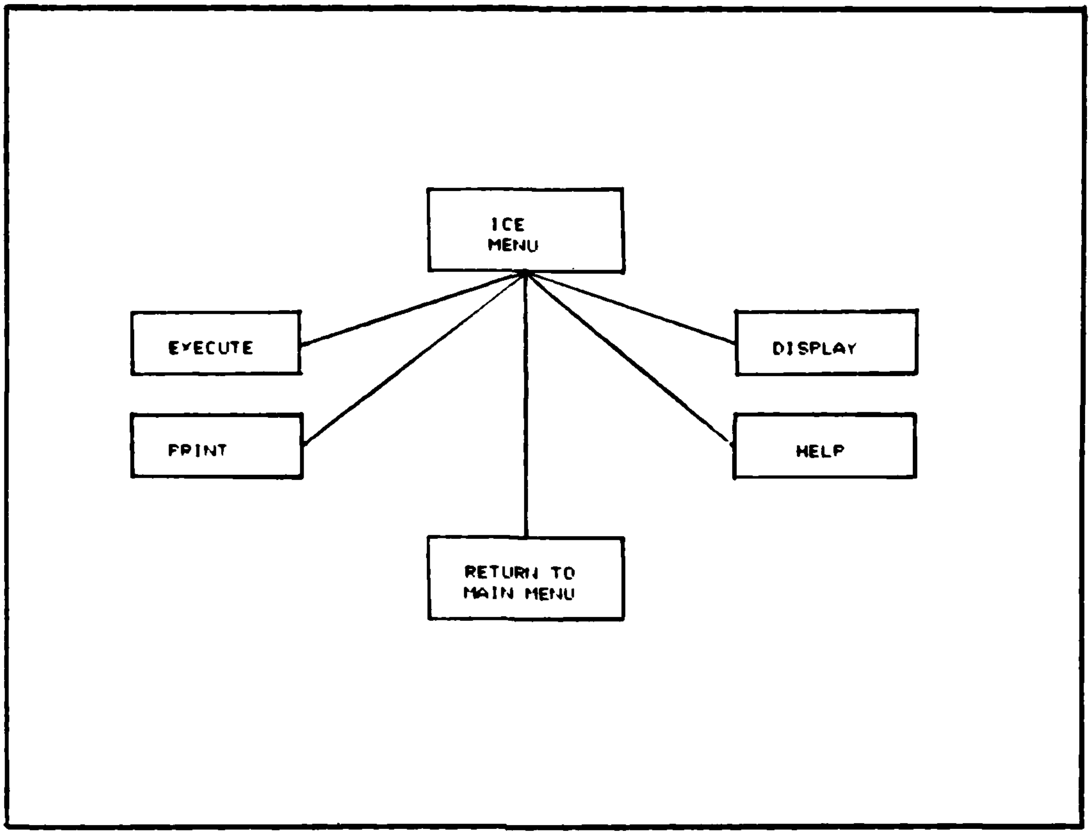


Figure 3. ICE Menu Options

Execute. The selection of this option resulted in the development of the "temp.txt" file. The "temp.ckt" file was configured for LOGSIM operation and therefore must be reformatted prior to use by ICE. The reformatted file was labeled "temp.txt." Once the file was reformatted the graph program was exited with all temporary files maintained. A supervisory program executes ICE (when ICE is available) and upon completion of the ICE execution, restarts the graph. The first test conducted by the graph program upon start up was for the existence of the "temp.txt" file. If it was present, the circuit was redrawn, based on the temporary circuit design files, and the ICE output is formatted for display and stored in the "temp.iot" file. The "temp.txt" file was then deleted.

View Rslt. The selection of this option resulted in the circuit design display being replaced by a text display of the ICE execution results. This was accomplished by displaying the contents of the "temp.iot" file. The user was also prompted to press any key when more than one screen worth of information was present or when ready to return to the circuit design screen.

Print. The selection of this option resulted in the user being prompted to ensure the printer was on-line and ready for printing. The "temp.iot" file was printed when the user followed the appropriate steps.

Help. The selection of this optirn resulted in the current screen display being replaced by a screen display containing a brief description of each of the "ICE" menu options. The user was prompted to press any key to continue. Once any key was depressed the "Help" screen display was replaced by the circuit design window.

Main Menu. The selection of this option resulted in the "Circuit Design" menu options being replaced by the "Main Menu" options.

LOGSIM Menu

The "LOGSIM Menu" was composed of seven options (see Figure 4).

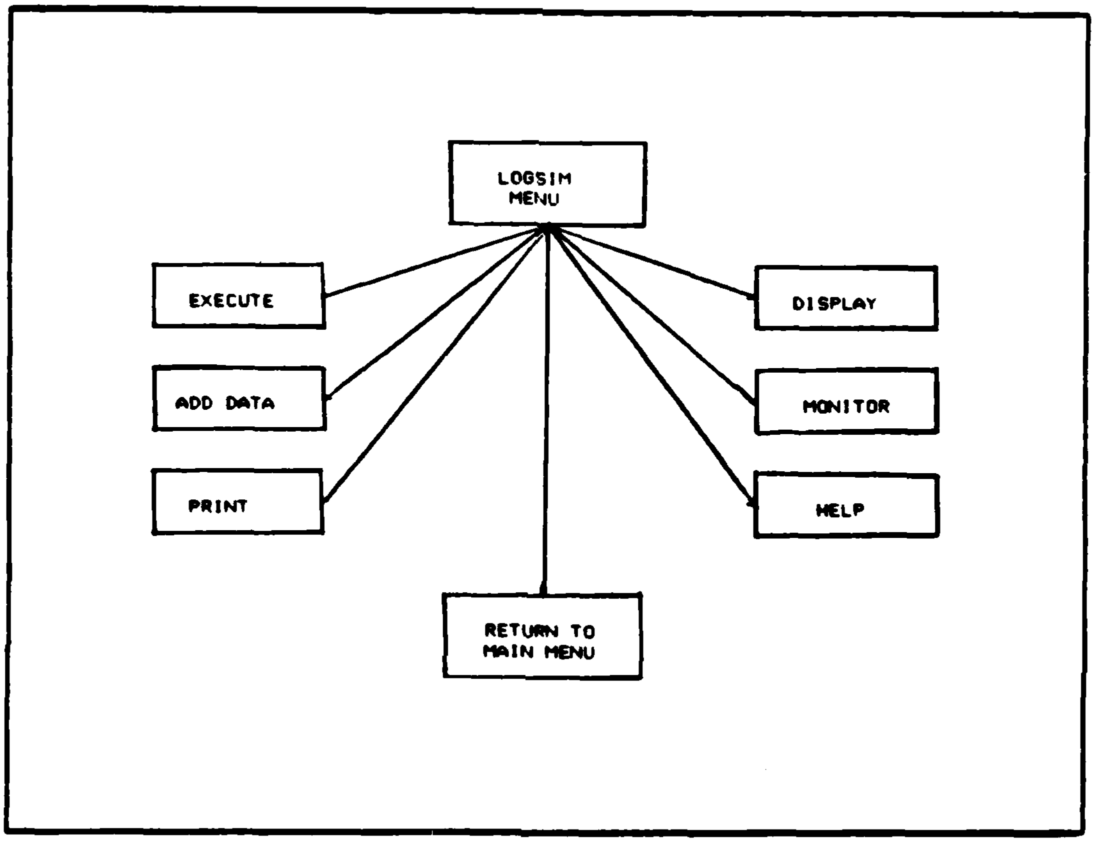


Figure 4. LOGSIM Menu Options

The first option, "Execute," provided the ability to execute the LOGSIM program. The "View Rslt" option provided the user the method of seeing the results of the LOGSIM execution. The "Add Inpts" option provided a means of allowing the user to specify an input data stream to the circuit. The "Monitor" option provided the user with the ability to specify locations where the data could be monitored. The "Print" option provided the user a means of getting a printout of the LOGSIM results. The "Help" option provided the user a recap of the function of each of the LOGSIM Menu options. The final option, "Main Menu," provided a means of returning to the previous menu.

Execute. The selection of this option resulted in the user being advised LOGSIM was in progress and a system call was made for the execution of LOGSIM. Once the LOGSIM program was finished, the user was prompted to press any key to continue.

View Rslt. The selection of this option resulted in the user being prompted for the type of output display format (i.e., a simulated wave form or a binary text). If the user entered a "W" or "w" the wave form option was selected; otherwise, the binary format was provided. The wave form format selection of this option resulted in the circuit design display being replaced by a mixture of text and graphics which provided a display of the decoded input data streams and the LOGSIM output file (which were stored in the "temp.wav" file). The wave form presentations were an output feature of the LOGSIM program. The user was also prompted to press any key when more than one screen worth of information was present or when ready to return to the circuit design screen. The binary format selection of this option resulted in the circuit design display being replaced by a text display of the decoded input data streams (which were stored in the "temp.ind" file) and the LOGSIM output file (which was stored in the "temp.out" file). The user was also prompted to press any key when more than one screen worth of information was present or when ready to return to the circuit design screen.

Add Inpts. This option was only applicable for designs in which external input to circuit designs is made (i.e., the circuit has at least one input port). The user was prompted to supply the period of LOGSIM simulation (i.e., the number of clock cycles, between 1 and 40, that the program was to execute). The number 40 was selected for the maximum clock cycle length because it allowed the input and output data streams for all ports and specified pin locations (identified using the "Monitor" option described below) to be displayed on the single horizontal line on the screen with a label. The label identified the specific data stream. Limiting a data stream to one horizontal line allowed the user to see the relationship between multiple data streams, each aligned and horizontally displayed, at a specific clock cycle by viewing the data vertically. If the user specified a number that was not in the range specified, an error message was posted and the option terminated. If the number was within the specified range, the user is prompted to provide the input data stream for each input port. The input ports are obtained from a review of the "temp.icn" file and the name was displayed. If the user provided an invalid input stream (e.g., numbers other than "0" or "1," or characters) an error message is posted and the user is prompted to try again. The data streams were then stored in the "temp.in" and "temp.ind" files. The "temp.in" file is used by LOGSIM and the "temp.ind" file is used for the "View Rslt" option. The difference between the files is the "temp.ind" file contains the two-character name for the input port. Once the data stream for the last input port has been inserted, the "Add Inpts" option is terminated. If the user added or deleted any input port, the user was required to redo this option, because the "temp.in" and "temp.ind" files were emptied.

Monitor. The user can only monitor signals located at pin locations on TTLs, ports, and the clock. The user was prompted to click the left mouse button at monitoring pin locations. If the user clicks the left button, a test is conducted to determine if the location is a pin. If the location is at a pin then the monitoring location is stored in the "temp.dis" file, unless the pin location is to the power or ground icons. If the location was anything else, an error message was posted. Once the user clicked the right mouse button the "temp.dis" file was closed and the "Monitor" option terminated. Each time this option was selected, the previous "temp.dis" file was overwritten.

Print. The selection of this option resulted in the user being prompted to ensure the printer was on-line and ready for printing. The "temp.ind" and "temp.out" files were printed when the user followed the appropriate steps.

Help. The selection of this option resulted in the current screen display being replaced by a screen display containing a brief description of each of the "LOGSIM" menu options. The user was prompted to press any key to continue. Once any key was depressed the "Help" screen display was replaced by the circuit design window.

Main Menu. The selection of this option resulted in the "Circuit Design" menu options being replaced by the "Main Menu" options.

File Description

The ability to design a circuit was worthless unless a method was available to save the information for later retrieval and review. Considerable review was accomplished to decide whether to use an existing Database Management System (DBMS) or to develop a set of flat files and the basic management (e.g., save, retrieve, delete) features. Two considerations, size of memory required and whether the software had to be purchased separately, were important in the decision as to the data storage technique. Both considerations were associated with the portability of the graphics interface program. Memory size was considered because the portability of a program is reduced if the user needed a machine with large available memory to accommodate the DBMS. The second consideration, DBMS procurement, introduced the issue of either supplying the DBMS with the program or requiring the user to supply a compatible DBMS. Since supplying a proprietary DBMS was not possible, a search of available public domain software was accomplished. The DBMS required the ability to be manipulated indirectly by an independent program (e.g., the graphics interface program) using the C language. The search for a public domain DBMS meeting these requirements was unsuccessful. This resulted in two possible options, specify a proprietary DBMS or develop a DBMS replacement. The decision was to develop the minimum features needed in a DBMS; thereby reducing DBMS operation overhead and precluding the need for the user to provide a proprietary DBMS.

To accomplish this effort, files were developed to store the circuit and the results of the LOGSIM and ICE execution. Until a user saves the circuit, all the data is stored in the temporary circuit design files. Once the user selected the "Save" option, a copy of all the files except "temp.txt," was made to the specified filename based on the appropriate extension listed below. A brief description of the files follows:

The "temp.loc" file contained the location data for the icons (i.e., power, clock, ground, TTLs, and ports). This file was used for preventing icon overlap during placement of TTLs or ports. The file consisted of one line entries specifying the X and Y coordinates of the reference point for the reserved space, the width and height of the space reserved for the icon, the code for the icon (i.e., 0 - TTL, 1 - Input Port, 2 - Power, 3 - Ground, and 4 - Clock), and the icons id number. The id number for Power, Ground and Clock icons was zero. For example, the entries for the Power icon, a possible TTL, and an Input port are shown in Figure 5.

```
   3       0      35      25    2    0
  59     124      90      81    0    1
 411      51      28      20    1    1
```

Figure 5. Example of three "temp.loc" File Entries

The "temp.icn" file contained a listing of all the icons in the circuit design and their characteristics (e.g., title, reference coordinates, color, and identification number). This file was also used for identifying the input ports for the LOGSIM "Add Inpts" option. The file consisted of one line entries specifying the type of icon, the id number, title, user specified X and Y coordinates, the systems assigned X and Y coordinate reference point, and the color. The available colors were numbered between 0 and 15. For example, the data which would correspond for the Power icon, the Input port and the TTL presented in the previous example is shown in Figure 6.

```
P 0   POWER   USER   0    0  SYSTEM   0   0  COLOR 0
T 1    7420   USER  68  124  SYSTEM  64 124  COLOR 7
I 1      EN   USER 420   51  SYSTEM 416  51  COLOR 4
```

Figure 6. Example of three "temp.icn" File Entries

The "temp.grf" file contained the necessary data to reconstruct the links of the circuit (i.e., path, link id, and color). The file consisted of three line entries per link. The first entry contained the X coordinate of the starting location, the ten turning points, and the ending point. The second entry contained the Y coordinate for the respective X coordinate. The third entry provided the color and link id (code) number. For example, the three entries for a possible link is shown in Figure 7. Note: If the user did not use ten turning points, the remaining turning point coordinates are set equal to the ending point.

```
X-COORD 10 200 220 134 148 148 148 148 148 148 148 148
Y-COORD 13  11 229 231 235 235 235 235 235 235 235 235
COLOR  14  CODE  52
```

Figure 7. Example of one "temp.grf" File Entry

The "temp.ckt" file contained a listing of the circuit interconnections (i.e., the starting pin location, link id, and ending pin location). This file was used by LOGSIM to establish the circuit design parameters. In addition, it was also used as the basis for the reformatted file (i.e., "temp.txt") described below. The file consisted of one line entries containing the information on the starting and ending component. The first half (i.e., the data prior to the ":") of the line entry corresponds to the component at the links starting end and the second half (i.e., the data after the ":") corresponds to the component at the other end of the link. For example, the connection established between the input port and the TTL , used in the previous examples, is shown in Figure 8. The first character in each part corresponds to the icon type (i.e., "I" for input, "T" for TTL, "P" for Power, "G" for Ground, and "C" for clock). Next, the number, corresponds to the icons id number. The next portion, corresponds to the title of the icon. The next number corresponds to the pin number. The pin number for an input port is always "1." The number that proceeds the "#"s in the first part is the link id number. Note: The example only displays a portion of the "#"s which were present. The first part actually contained sixteen "#"s and the second part contained twenty "#"s. The "#"s are used to reserve space for comments produced/needed by either the ICE or LOGSIM programs.

```
I 1   EN  0  451**********:T 1 7420 10 **********
```

Figure 8. Example of one "temp.ckt" File Entry

The "temp.txt" file contained the reformatted information contained in the "temp.ckt" file. This file provided no additional data and therefore is never maintained during the "Save" operation. The file was also used as a flag for determining if the graph program is initially starting up (if the file was not present) or restarting after ICE execution (if the file was present). (See Figure 9). Note: Each part actually contained twenty "*"s.

```
I1:EN:0:**********:T1:7420:10:**********
```

Figure 9. Example of one "temp.txt" File Entry

The "temp.iot" file contained the ICE output data. This file was a consolidated file of the questionable or missing links as determined by the ICE execution. This file was used for the "View Rslt" and "Print" options in the ICE Menu. The file contained user readable text.

The "temp.in" file contained the input data stream for the each of the external input ports. This file is used only by LOGSIM. The file contained one line entries for each of the input ports, consisting of the input port id number followed by a semicolon and then the input data stream. (See Figure 10).

```
001:01010101010101010101
```

Figure 10. Example of one "temp.in" File Entry

The "temp.ind" file contained the same information as the "temp.in" file but included the name of the input port and twenty "*"s for display alignment. This file is used for the "View Rslt" and "Print" options in LOGSIM Menu. (See Figure 11).

```
EN = 001    ********************:01010101010101010101
```

Figure 11. Example of one "temp.ind" File Entry

The "temp.dis" file contained the necessary data (e.g., the TTL or port identification number and pin number) on the pin locations where the user wants LOGSIM monitoring data. The file consisted of one line entries for each monitoring point. The entry contained the icon type, id number, title, and pin number and some "*"s for possible comments. For example, if the user requested to monitor pin 2 on the 7420 described earlier the file would contain the entry shown in Figure 12.

```
T  1  7420 2********************:
```

Figure 12. Example of one "temp.dis" File Entry

The "temp.out" file contained the binary formatted output data from LOGSIM. This file is used for the "View Rslt" and "Print" options in the LOGSIM Menu. (See Figure 13).

```
IC # 1 (SN  7420) PIN # 2  :1010101101010101010110
```

Figure 13. Example of one "temp.out" File Entry

The "temp.wav" file contained the wave form formatted input and output data from LOGSIM. This file is used only for the "View Rslt" option in the LOGSIM menu. The "0"s and "1"s in the previous example would be replace by graphical substitutes (i.e., dashes for "0"s and small boxes for the "1"s).

Summary

An extensive review of the detailed design used in the development of the graphics interface program was presented. The program was divided up into four major parts, corresponding to each of the menus. The Main Menu provided the basic tools necessary to manage the overall circuit design environment. The Circuit Design Menu provided the tools to create and modify current circuit design. The ICE Menu provided the interface to the circuit evaluation program. Finally, the LOGSIM Menu provided the interface to the circuit simulation program. Furthermore, this chapter included a comprehensive recap of the underlying files which established the basic structure of the program and the interface to the other digital design tools.

V. Implementation

Implementation

The graphics program, developed by Bruce Clay (9), provided the underlying graphic primitives (e.g., drawing lines, rectangles, and displaying text on the screen) and the general format for portions of the graphics interface program. For example, the drawings of TTLs and input ports were accomplished by use of the box drawing feature of Clay's program. In addition, the menu format and the function of switching between menus were also from Clay's graphics program. Furthermore, portions of the code for the graphics interface program were incorporated in the code for Clay's graphics program. However, the graphical display was only a small function in the overall interface program. The crucial part of the interface program was the interpretation, storage, retrieval and modification of the graphically displayed digital circuit. As a digital circuit was designed, the user specified information was interpreted and formatted for future ICE and LOGSIM implementation. Several meetings were conducted with W. Deloria (1) and S. M. Wagner (2) to establish a format for their input/output files. Once those files were established the three thesis efforts became completely independent. Integration of ICE and LOGSIM with the graphics interface was accomplished using "system" calls and the temporary circuit design files described in the previous chapter. These files were created based on the requirements of the destination program, prior to the program's execution, and then a "system" call was made to the appropriate program. Upon termination of the ICE or LOGSIM program, the output files were formatted by the graphics interface program for display/printing. The overall implementation of the principles described in the previous chapter was successful; however, minor problems were encountered.

Problem Areas

Minor modifications to the overall methodology for integrating ICE, LOGSIM and the graphics interface were required. The first minor change incorporated the development of a supervisory program which initiated the graphics interface program and upon its termination, implemented the ICE program if the "temp.txt" file was present, otherwise termination of the supervisory program was accomplished. If the ICE program was initiated, than upon its termination, the graphics interface program was restarted. The supervisory program was named IDIET, an acronym for Integrated Digital Engineering Tool. The relationship between the programs (i.e., IDIET, ICE, LOGSIM, and the graphics interface) and the database is shown in Figure 14. The need for this approach was necessary because the operation of the ICE program incorporated two features which conflicted with the graphics interface. The primary conflict was the need for memory space which the graphics

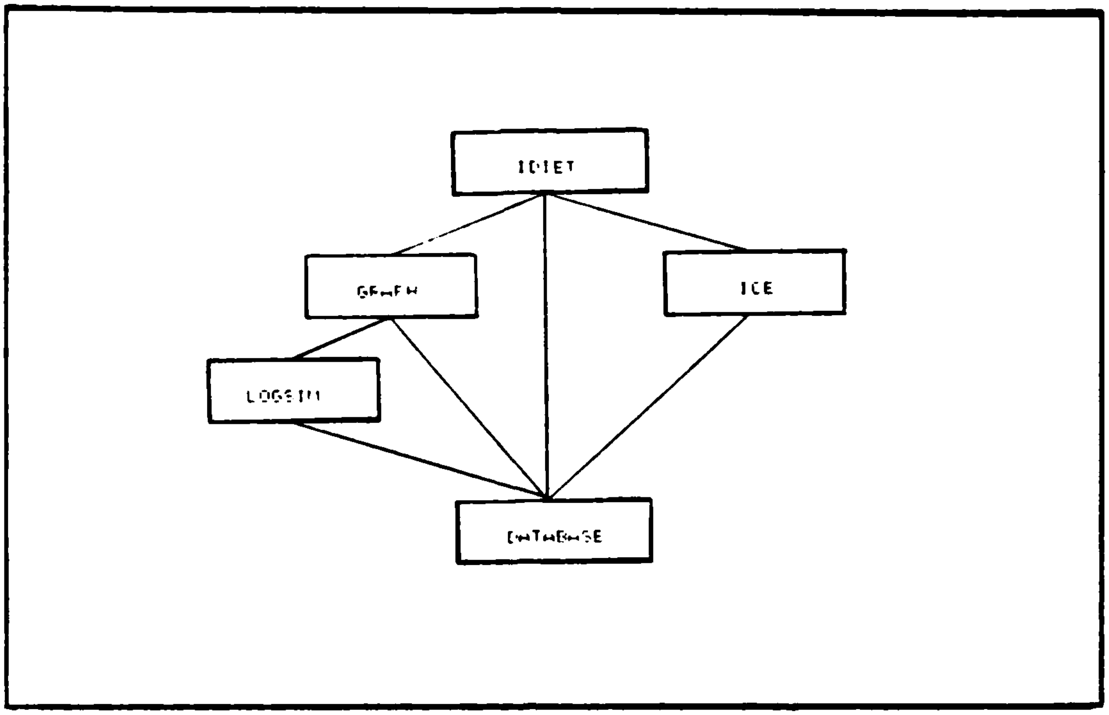


Figure 14. Integration Architecture

interface program was using. The other conflict was that the ICE program also incorporated graphical displays on the screen which corrupted the circuit design screen of the graphics interface package. The ultimate solution was to terminate the graphics interface package prior to ICE implementation. Using this approach made it necessary to develop a means of detecting upon start up of the graphics interface program whether it was an initial start-up or a return from an ICE execution. The graphics interface, upon start up, tests for the "temp.txt" and if it is present, returns the user to the appropriate menu and redraws the circuit. If "temp.txt" is not present, the program opens with the "Main Menu." When the user selected the "Exit" option, all the temporary circuit design files (including "temp.txt") were deleted and the graphics interface program terminated. Upon termination of the graphics interface program the supervisory program tested for the existence of the "temp.txt" file and since it was not present the supervisory program also terminated.

Two commercially available programs (i.e., mouse driver and graphics screen dump) must be procured for the graphics interface package to be complete. The mouse driver used to support this thesis was "MSMOUSE.COM," version 5.03 (manufactured by Mouse Systems Corp). The mouse driver was essential for the operation of Bruce Clay's program and the graphics interface program. The graphics screen dump was not essential for the program operation. However, a graphics screen dump program provided the only means for the user to obtain a printout of the graphical display of the circuit design. The "Print" option in the ICE and LOGSIM menus did not use a graphics screen dump approach, but used the print feature provided by the host operating system.

Summary

The merging of Bruce Clay's graphics program and the graphics interface program was successful. No problems were noted in the integration of LOGSIM and the graphics interface; however, minor changes in the execution of ICE from the graphics interface program was necessary. The changes were made to overcome the conflicts of available memory and graphics corruption. The requirement to obtain the mouse driver and graphics screen dump programs to complete the graphics interface package was a noted limitation.

VI. Testing and Results

Testing

This chapter outlines the test set used to evaluate the graphics-oriented interface program and recap the results obtained. The test set was not developed to validate the ICE or LOGSIM programs, just the interface into them. Each module of the interface program was evaluated using the "white box testing" (10:472) approach, during its development. This entailed testing each possible user specified option or program execution path. In most cases, an invalid input generated an appropriate error; however, some errors resulted in operating system responses, which could not be resolved prior to error execution. For example, an operating system error message resulted if the user specified an invalid disk drive in the filename when using the "Save" option (e.g., using filename "E:TEST" for a system which did not have an "E" disk drive). The remaining emphasis of the evaluation of the program was in "black box testing" (10:484).

To develop a reasonable number of tests which could evaluate all the possible circuit designs for the limited library of 32 TTLs was impossible. This is due to the unlimited positioning of TTLs on the screen and the unlimited number of link path possibilities given fixed TTL locations. In addition, the user had the capability to add a virtually unrestricted number of input ports (only restricted by "temp.icn" file size or space on the screen). Therefore, the test set was designed to conduct an extensive evaluation on each of the menu options. A brief description of the types of tests conducted on each of the menu options in all the menus follows.

Main Menu Options. The test sets for the options which were available on the "Main Menu" follows:

Design Circuit. The test for this option was to determine if selecting this option resulted in a replacement of the "Main Menu" by the "Circuit Design" menu.

Retrieve Circuit Option. The test set included: attempting to retrieve a circuit while another was already in progress, attempting to retrieve a circuit which did not exist, retrieving a previously saved circuit from the default disk drive (i.e., the hard disk), and retrieving a previously saved circuit from a user specified disk drive (i.e., the floppy disk).

Delete Circuit Option. The test set included: Deleting the circuit design which was in progress, and deleting when then was no circuit in progress.

Save Circuit Option. The test set included: Attempting to save when no circuit was in progress, saving the current circuit design to the default disk drive (i.e., the hard disk), and saving the circuit to a user specified disk drive (i.e., the floppy disk). In addition, a valid and an invalid filename was used in attempting to save the circuit to the default and specified disk drives.

ICE Menu Option. The test for this option was to determine if selecting this option resulted in a replacement of the "Main Menu" by the "ICE Menu". This option would result in an error message if the ICE program was not available or if a circuit was not in progress.

LOGSIM Menu Option. The test for this option was to determine if selecting this option resulted in a replacement of the "Main Menu" by the "LOGSIM Menu". This option would result in an error message if the LOGSIM program was not available or if a circuit was not in progress.

Help Option. The test for this option was to determine if the current screen display was replaced by a screen display containing the information contained in the "HELPO.DOC" file (HELP information for Main Menu). The test also included the pressing of a key while the help screen was displayed, to see if it would restore the circuit design screen display and also the Main Menu options.

Exit Option. The test for this option was to determine if the user would receive an exit warning prompt, the answer to which determined if the program was terminated. The test included specifying a valid and an invalid termination request. If the user specified a valid termination request (i.e., entering a "Y" or "y" to the prompt), all the temporary circuit design files were deleted prior to termination.

Circuit Design Menu Options. The test sets for the options which were available on the "Circuit Design" Menu follows:

Add TTL Option. The test set for this option included: reviewing the possible TTL chips display (i.e., TTLs listed in the "TTL.DES" file), attempting to add invalid TTLs, attempting to add valid TTLs at invalid locations (e.g., where the TTL would overlap existing TTLs, ports, power icon, ground icon, clock icon, or circuit design window border), and adding each valid TTL in at least one valid location. Once the TTL was added, a review of the "temp.loc" and "temp.icn" files would confirm the TTL was properly stored. In addition, the graphical display of the TTL would reflect the appropriate title (e.g., 7400) and the next sequential TTL id number. (Note: TTL id numbers are sequentially assigned, based on the highest existing number. TTL id numbers of chips deleted are not replenished, unless the TTL deleted had the highest TTL id number.)

Delete TTL Option. To delete a TTL the user must click on any pin of the desired TTL and the TTL must not have any existing links connected. The test set for this option included: attempting to delete TTLs with existing links, attempting to delete something other than a TTL (e.g., ports, power icon, ground icon, or clock icon), and deleting a TTL with no existing links. Error messages advise unsuccessful deletion attempts. A successful deletion results in the screen display being redrawn without the specified TTL. Once the TTL was deleted (if allowed to delete) a review of the "temp.icn" and "temp.loc" files would confirm the TTL deletion. (Note: these files are the primary means used to draw the TTL and are also used for establishing the valid link connection points.)

Add Link Option. Links can only be added at existing pin locations, can have at most ten bends (turning points), must remain within the circuit design window, and must also terminate at an existing pin location. The test set for this option included: attempting to start or end a link at an invalid location, attempting to exceed the ten turning points restriction, attempting to draw a link outside of the design window, and drawing a valid link. Error messages advise the user of invalid actions. A successful link addition would result in the link path, color and link id number being stored in the "temp.grf" file. In addition, a listing in the "temp.ckt" file would indicate an established circuit connection.

Delete Link Option. Links can only be deleted by specifying both of its termination pins. The test set for this option included: attempting to specify invalid pin locations (i.e., specifying something other than a pin location), attempting to delete a nonexistent link (i.e., specifying two valid pin locations with no link between them), deleting a valid link. Error messages advise the user of invalid pin locations and nonexistent links. A successful link deletion results in the screen being redrawn without the specified link. In addition, the "temp.grf" and "temp.ckt" files no longer reference the link or circuit connection.

Add Port Option. The test set for this option included: attempting to specify ports with an invalid name (e.g., name not containing two characters), attempting to add valid ports at invalid locations (e.g., where the port would overlap existing TTLs, ports, power icon, ground icon, clock icon, or circuit design window border), adding a valid port in a valid location. Once the port was added a review of the "temp.loc" and "temp.icn" files would confirm the port was properly stored. In addition, the graphical display of the port would reflect the appropriate title (e.g., I1). The next sequential port id number would be available in the files mentioned. (Note: port id numbers are sequentially assigned, based on the highest existing number. Port id numbers of ports deleted are not replenished, unless the port deleted had the highest port id number.)

Delete Port Option. To delete a port the user must click on the pin of the desired port and the port must not have any existing links connected. The test set for this option included: attempting to delete ports with existing links, attempting to delete something other than a port (e.g., TTLs, power icon, ground icon, or clock icon), and deleting a port with no existing links. Error messages advise unsuccessful deletion attempts. A successful deletion results in the screen display being redrawn without the specified port. Once the port was deleted (if allowed to delete) a review of the "temp.icn" and "temp.loc" files would confirm the port deletion. (Note: these files are the primary means used to draw the port and are also used for establishing the valid link connection points.)

Set Color Option. The test of this option was to determine if the user could change the color of subsequent screen images (i.e., TTLs, ports, and links).

Help Option. The test for this option was to determine if the current screen display was replaced by a screen display containing the information contained in the "HELP1.DOC" file (HELP information for Circuit Design Menu). The test also included the pressing of a key while the help screen was displayed, to see if it would restore the circuit design screen display and the Circuit Design Menu options.

Main Menu Option. The test for this option was to determine if selecting this option resulted in a replacement of the "Circuit Design" menu by the "Main Menu."

ICE Menu Options. The test sets for the options which were available on the "ICE" Menu follows:

Execute Option. The test for this option was to determine if the selection of this option resulted in the creation of the "temp.txt" file, the termination of the graphics interface program and the start up of the ICE program. Once the ICE program was terminated the graphics interface program was restarted and the user was returned to the screen display prior to ICE execution time and the ICE output files were formatted in the "temp.iot" file.

View Results Option. The test for this option was to determine if the current screen display was replaced by a screen display containing the information contained in the "temp.iot" file (output information from ICE). The test also included the pressing of a key while the result screen was displayed, to see if it would page through the remaining result displays and then restore the circuit design screen display and the ICE Menu options.

Print Option. The test for this option was to determine if a hard copy of the "temp.iot" file could be obtained if the user followed the instructions provided by the prompts. The printout should be blank in the cases where ICE had not yet been executed.

Help Option. The test for this option was to determine if the current screen display was replaced by a screen display containing the information contained in the "HELP2.DOC" file (HELP information for ICE Menu). The test also included the pressing of a key while the help screen was displayed, to see if it would restore the circuit design screen display and the ICE Menu options.

Main Menu Option. The test for this option was to determine if selecting this option resulted in a replacement of the "ICE" Menu by the "Main Menu."

LOGSIM Menu Options. The test sets for the options which were available on the "LOGSIM" Menu follows:

Execute Option. The test for this option was to determine if the selection of this option resulted in the user being advised that LOGSIM was initiated. Once the LOGSIM program was terminated the graphics interface program advised the user that LOGSIM was finished.

View Results Option. The test for this option was to determine if the current screen display was replaced by a screen display containing the information contained in the "temp.ind" and "temp.out" files (input data stream and output information from LOGSIM). The test also included the pressing of a key while the result screen was displayed, to see if it would page through the remaining result displays and then restore the circuit design screen display and the LOGSIM Menu options.

Add Inputs Option. The test for this option included: specifying an invalid period of execution (i.e., clock cycles not between 1 and 40), invalid number of inputs per input port (e.g., inputting five bits of data when six clock cycles was specified), and invalid data format (i.e., values other than "0" or "1"). In addition, a combination of these tests were also accomplished (e.g., invalid number and format of input data stream). A review of the "temp.in" and "temp.ind" files provided a means to determine if the resultant input stream was properly formatted for LOGSIM execution and visual display (using the LOGSIM "View Rslts" option).

Monitor Option. Monitoring could only be accomplished at pin locations: therefore, tests were made to attempt to establish monitoring at both valid and invalid locations. Error messages were expected for invalid user specified locations. A review of the "temp.dis" file provided a means to determine if the monitoring points specified were properly formatted for LOGSIM execution.

Print Option. The test for this option was to determine if a hard copy of the "temp.ind" and "temp.out" files could be obtained if the user followed the instructions provided by the prompts. The printout should be blank if the user did not specify any input data (i.e., did not select "Add Inpts" option) and did not execute LOGSIM. If LOGSIM was executed prior to the "Print" option selection, the "temp.out" file as a minimum would be printed. It is possible that the circuit not have any external input ports required; therefore, no "temp.ind" file would exist.

Help Option. The test for this option was to determine if the current screen display was replaced by a screen display containing the information contained in the "HELP3.DOC" file (HELP information for LOGSIM Menu). The test also included the pressing of a key while the help screen was displayed, to see if it would restore the circuit design screen display and the LOGSIM Menu options.

Main Menu Option. The test for this option was to determine if selecting this option resulted in a replacement of the "LOGSIM" menu by the "Main Menu."

Results

A comprehensive series of tests, which included all the test sets described above, was conducted using the graphics interface program and its tools. Minor errors surfaced during the test which were corrected immediately. The unexpected errors included: misspelled warning/prompt messages and mouse related deficiencies. For example, if the user was adding a link and pressed the left mouse button at a valid starting position (i.e., an existing pin location) but moved the mouse prior to releasing the button, the link was drawn, starting at the position where the cursor was when the button was released. This mouse/program code deficiency was overcome with a minor change in the program code.

The ability to construct a valid circuit design display was demonstrated and invalid user inputs resulted in the appropriate error warning message being posted. The ability to save and retrieve circuit designs and results of ICE and LOGSIM executions was also tested and found successful. The circuit design files, which provide the data storage and retrieval method, were extensively reviewed and determined to be accurate.

Users not following the guidance provided in the user's manual (see appendix) or the program's user prompts (i.e., messages located on the bottom of the screen) could receive corrupted screen displays (e.g., operating system error messages as described earlier) or program termination. Note: A user could recover if the screen became corrupted. To recover, the user saved the existing circuit to a selected file, the user then deleted the current circuit, and then retrieved the specified file. This resulted in the circuit being redrawn without the previous display corruption. However, circuit design data which was not saved prior to a program termination was lost.

Summary

The purpose of this chapter was to provide a general description of the test sets required to validate the program. Program modules where thoroughly evaluated prior to integration. Once the modules were integrated, a comprehensive set of tests where developed to evaluate each of the menu options. The test results were successful and the minor errors which arose where corrected.

VII. Recommendations and Conclusion

The aim of this thesis was to design and develop a graphics-oriented circuit design environment on a Zenith Z-248 workstation. The approach included researching existing methods for data management, using human-computer interface considerations to streamline input requirements and identifying the basic tools necessary to create a circuit design environment. The basic graphics tools (e.g., displaying lines, boxes, and text) were provided by the graphics program, developed by Bruce Clay. Clay's package provided the underlying structure and tools necessary for the construction of the graphics oriented user interface to the ICE and LOGSIM programs. This computer aided design tool provides the students in the undergraduate digital design lab with the capability to enhance their education by providing a means to reduce unproductive time spent trouble shooting circuits. In particular, this tool allows the user to develop circuit designs which can be evaluated by ICE, simulated by LOGSIM, and modified when needed. Once the user has received successful results from these programs, the circuit can be constructed using actual components.

Recommendations

The graphics oriented interface established the basic tools needed by the user to interface with the ICE and LOGSIM programs and to provide the user with a display of the proposed circuit design. However, these tools are only the building blocks for the tools that are yet needed to provide a comprehensive circuit design environment. The possibilities for future studies are endless; however, the following suggestions are encouraged:

First, the development of an expert system which will "optimize" the current design. The word "optimize" was selected to refer to the process of positioning the TTLs and ports in such a way as to reduce the number of links which overlap when the circuit is drawn. The feasibility to optimize circuit paths is demonstrated by commercially available programs which are used in the development of circuit boards.

Second, the ability to "ZOOM" or "SCROLL" on the current circuit design window. This feature is critical because the current circuit design window precludes the development of large circuits. In addition, small circuit designs which have numerous links tend to result in many links overlapping each other. The overlapping of the links can result in the user not being able to trace a link path. The development of zooming or scrolling would allow for better displays and ease of link path tracing. Note: The use of the term overlapping here refers to sharing a portion of a link's path (i.e., line segment), and not just a single point (i.e., crossing the path).

Third, the ability for the graphics interface tool to further process the output from the ICE program. The graphics interface program currently displays the ICE output as text. The ability for the program to provide the user with an option as to how the result is to be provided could be offered. For example, an option may be to cause links, which the ICE program identified as questionable, to blink. In addition, the pins, where ICE specified missing links exist, could be highlighted. These graphical options would make it easier and faster to isolate these links and pin locations, then just reading text.

Fourth, the ability for the graphics interface tool to further process the output from the LOGSIM program. The graphics interface program currently prompts the user to specify if a wave form or binary display is desired. Additional display options could be made available. For example, allow the user to specify labels for the monitoring points or reformat the output displays per the users needs.

Fifth, the current interface program uses an existing library of thirty-two TTLs. This library is limited as a result of the TTLs currently recognized by ICE and LOGSIM. A separate menu option or program should be developed to expand the library and provide the means for the ICE and LOGSIM programs to use this library. Currently, the interface program's library only specifies the title (e.g., 7400), description, and the number of pins. A consolidated library would require additional information about the TTLS. For example, the characteristics of the pins (e.g., pin 1 on a 7400 is an input to an "NAND" gate) would be required.

Sixth, the development of support programs to allow the graphics interface program to be completely portable. The interface required the use of two commercial programs, a mouse driver and a graphics screen dump program. Users were required to provide their own programs to support the interface program.

Seventh, the development of a Database Management System (DBMS) which is specifically designed to support this package. The current program used a flat file design for data storage. This approach was used because the program needed to be portable, free from license, and time precluded development of an appropriate DBMS. The development of this DBMS should increase the speed and reduce the duplication of data which occurred when using the flat file data storage approach.

Finally, a hierarchical or modular design approach (i.e., merging multiple digital circuit designs) should be implemented. This capability would allow for individual circuits to be developed independently and merged at a later time. This feature is possible if an external output port is created. The user can specify the linking of these external output ports to the input ports of existing circuits. These links would need to be established in a separate file which merges the files prior to ICE or LOGSIM execution.

Conclusion

The need for a graphics oriented interface tool which provided the user with an easy to use doorway into the ICE and LOGSIM programs was developed. The ICE and LOGSIM programs, tools themselves, provided the user with valuable information; however, they required the user to already have a circuit layout (e.g., drawn on paper). In addition, the format for both of these programs was not identical; therefore, the user had to make modification to the input files in order to use either program. The interface program allows the user to design the circuit on the screen and the program formats the required information for the ICE and LOGSIM programs. The interface program provided the basic tools needed to interface with the ICE and LOGSIM programs and provided the user a means of making circuit modifications. A comprehensive user's manual was also developed and is included as an appendix to this thesis effort. The program was evaluated using a comprehensive set of tests, which included all the test sets described earlier, and passed. However, the interface program introduced a size constraint on the design circuit, due to the absence of a zoom or scrolling feature. In addition, the requirement for the user to provide the commercial mouse driver and graphics screen dump programs were a limitation of Bruce Clay's graphics program which provided the underlying graphics tools.

Appendix:

User's Manual for the Graphics Oriented LOGSIM/ICE Interface

Written by: Capt Charles A. Adams Jr.

Date: December 1987

Table of Contents

Page

Introduction

General. This manual describes the basic procedures necessary to design digital circuits using the graphics oriented interface to the Logic Simulator (LOGSIM) and the Interconnect Expert (ICE). This graphics interface uses a mouse. "Clicking" is a basic term associated with using a mouse and it refers to pressing one of the buttons on the mouse. Generally the user positions the cursor (e.g., an arrow or cross hairs) at a selected location on the screen and then "clicks" the mouse. The middle button of a three button mouse provides the user a repeat option capability which is not available with only two button mice. For example, if the user has just completed adding a link to a circuit, clicking the middle button results in the same action as if the user pointed to the "Add Link" option and then clicked the left mouse button.

System Requirements. The program requires a Zenith Z-248 workstation (or compatible) containing an "EGA" card (with 256K memory) and a mouse driver (e.g., MSMOUSE.COM version 5.03 produced by Mouse Systems Corporation). Several types of mice designs exists. These designs vary in the number of buttons (i.e., 1, 2, or 3) to the method of implementing cursor movement (e.g., rolling ball, or optical). This program requires a mouse with a minimum of two buttons; however, a three button mouse is preferred. Furthermore, the program uses extensive file manipulation and a hard drive should increase the program's response. In addition, storage of large digital designs may require more memory than one floppy can store. However, a hard drive is not required.

Software installation. The program requires several files to be co-located in the same directory for its proper execution. These files include: HELP0.DOC, HELP1.DOC, HELP2.DOC, HELP3.DOC, TTL.DAT, TTL.DES, and GRAPH.EXE. In addition, LOGSIM.EXE is required for performing circuit simulations. The directory should also contain the Expert System, executable code and its associated files, which provides the circuit evaluation capability of ICE. Furthermore, the directory containing these files should have no files with a filename prefix of "TEMP", because the program during execution will possibly overwrite and then delete them.

Initial start-up. Immediately prior to executing the graph.exe file the user must activate the mouse driver. This is accomplished in several ways depending on the mouse. It is recommended a batch program be developed to accomplish the mouse and graph program start up routine. In addition, if a graphics screen dump capability is available, it should also be included in the batch program prior to executing "graph". Once the user initially starts up the program (or after a circuit has been deleted) a circuit must be created or retrieved. To create a circuit the user follows the

"Designing a circuit" procedure. To retrieve an existing circuit the user should know the filename of the circuit and follow the "Retrieving a circuit" procedure.

Human/computer communication. Throughout the operation of this program, communication between the user and the computer program is conducted via the communications line on the screen. The communications line is located at the very bottom of the screen (below the horizontal line of the graphics window). The program will prompt the user for input data or provide the user with information about the status of the program's circuit design process. It is imperative the user follows the instructions on the communications line, to ensure a proper functioning of the program.

Test Vector Generation. The user can add external input data streams (i.e., a series of "1"s and "0"s) to test the operation of a circuit by using input ports. These input ports provide the only means of external input to the circuit. Once an input port has been established, the user specifies the input data using the "Add Inpts" option in the "LOGSIM Menu". Furthermore, the only method of tracking the external output or status at any location in the circuit is via the monitoring option also provided in the "LOGSIM" menu.

Current Limitations. This tool currently has the following limitations:

1. Designs are restricted to the window size (i.e., no scrolling or zoom feature present to expand circuit size).

2. Only 32 TTL chips are available in the design library.

3. The user can only obtain a graphical output of the circuit by using a commercially available tool.

I. Menu Descriptions:

This graphics oriented tool for the LOGSIM/ICE programs is implemented by four menus: Main Menu, Circuit Design Menu, ICE Menu, and LOGSIM Menu. A general description of each of these menus follows:

Main Menu. This menu provides the user with the basic functions necessary in the maintenance of a circuit design. (See Figure A-1). The remaining three menus are accessed from this menu and return only to this menu. Finally, termination of the program can only occur from this menu.

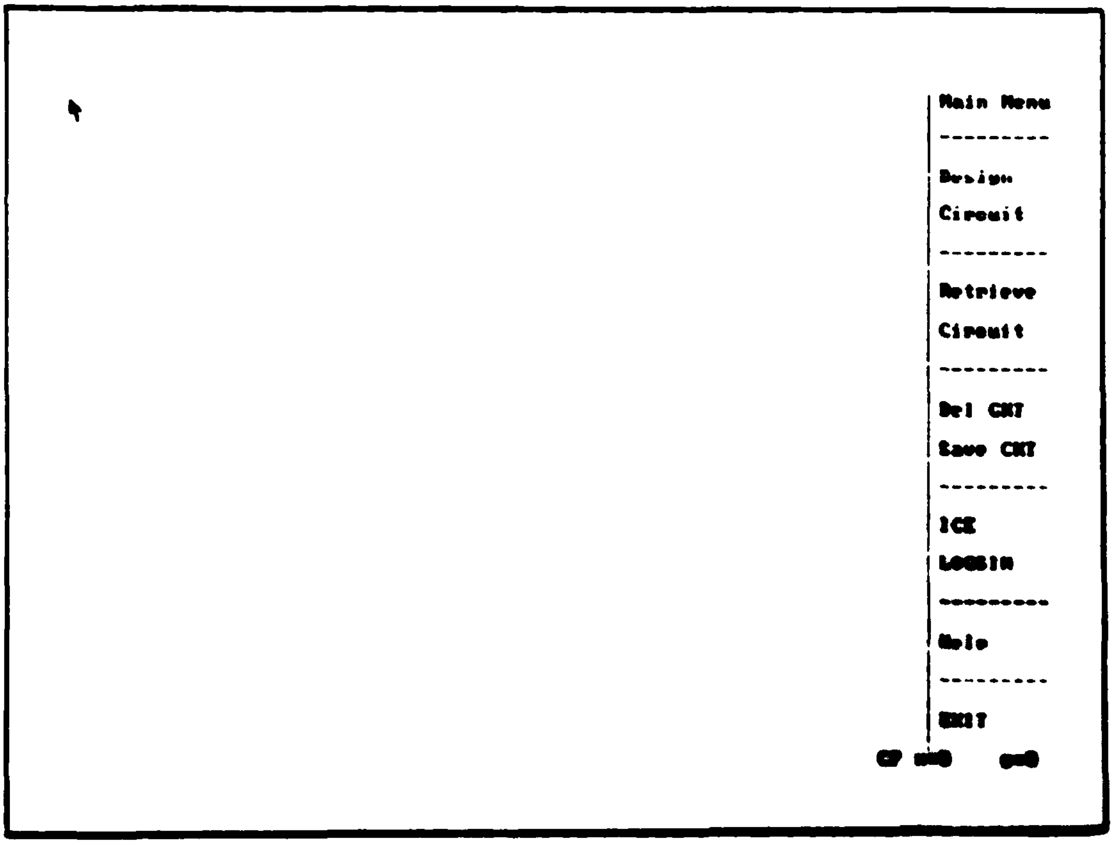
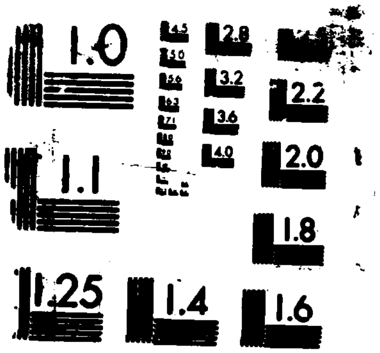


Figure A-1. Display of "Main Menu"

AD-A188 831 A DIGITAL CIRCUIT DESIGN ENVIRONMENT(U) AIR FORCE INST 2/2
OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING
C A ADAMS DEC 87 AFIT/GCS/ENG/87D-1

UNCLASSIFIED F/G 9/1 NL



MICROCOPY RESOLUTION TEST CHART

Circuit Design Menu. This menu provides the user with all the necessary tools to draw a digital circuit design on the screen. In addition, from this menu the user can alter existing circuit designs, select the color of screen displays and return to the "Main Menu". (See Figure A-2).

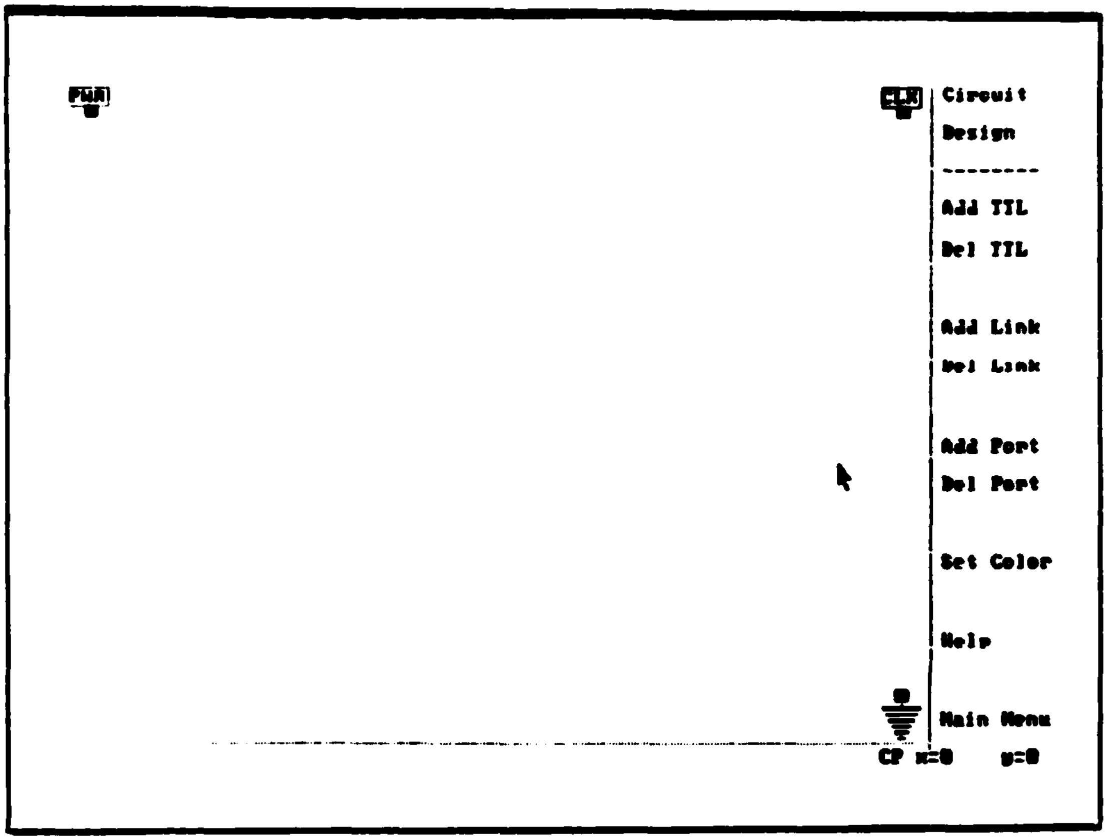


Figure A-2. Display of "Circuit Design Menu"

ICE Menu. This menu provides the user with an interface to the ICE program. A circuit, previously designed using the "Circuit Design Menu" options, can be tested using ICE and the results reviewed using the options in this menu. (See

Figure A-3). This menu is unavailable to the user until a circuit has been designed or retrieved.

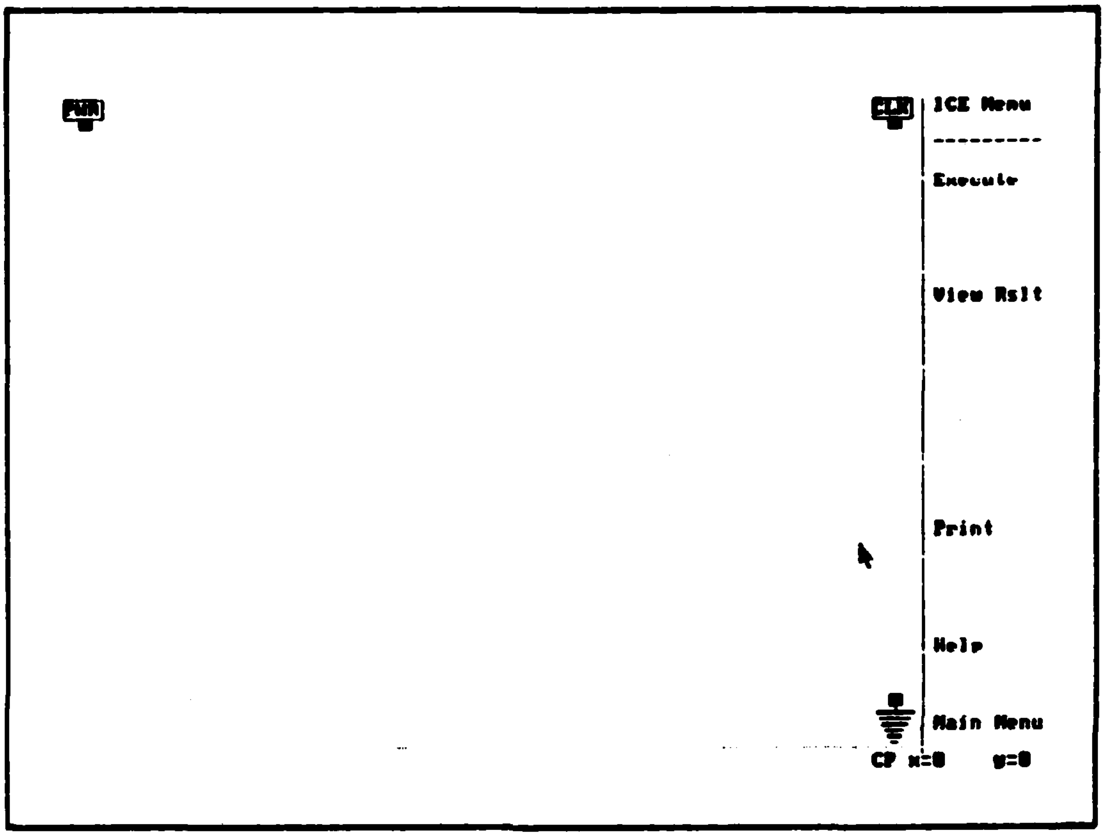


Figure A-3. Display of "ICE Menu"

<u>LOGSIM</u> <u>Menu.</u> This menu provides the user with an interface to the LOGSIM program. Input data stream(s) and monitoring locations can be specified for the circuit. Furthermore, the circuit can be simulated using LOGSIM and the results reviewed. (See Figure A-4). This menu is unavailable to the user until a circuit has been designed or retrieved.

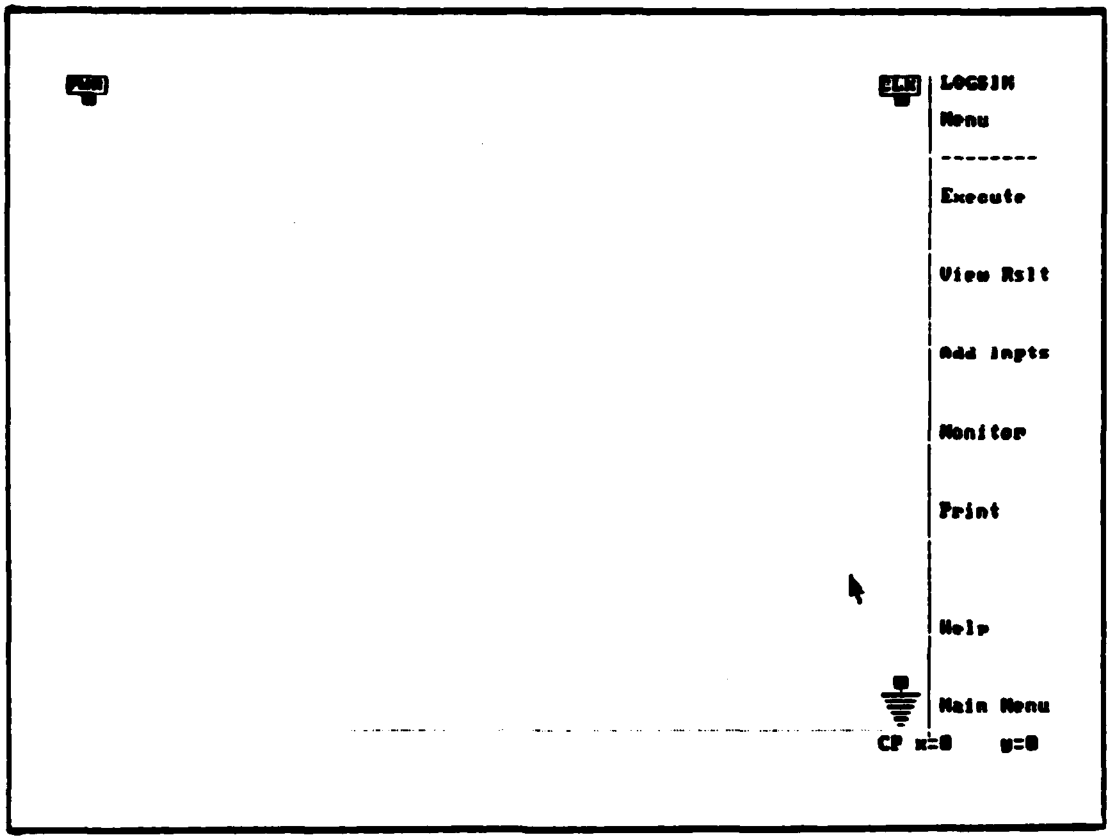


Figure A-4. Display of "LOGSIM Menu"

II. Designing a circuit:

To create a new circuit or make modifications to the circuit currently in memory, the user clicks on the "Design Circuit" option in the "Main Menu". (See Figure A-5).

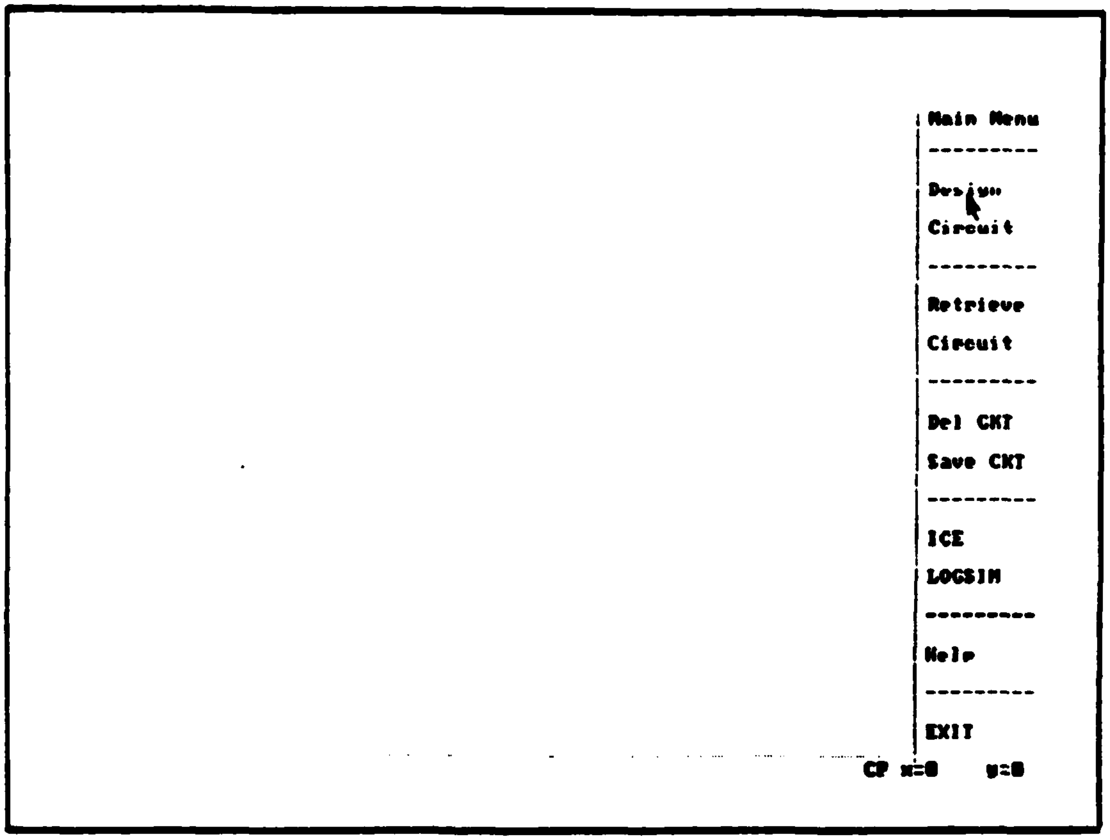
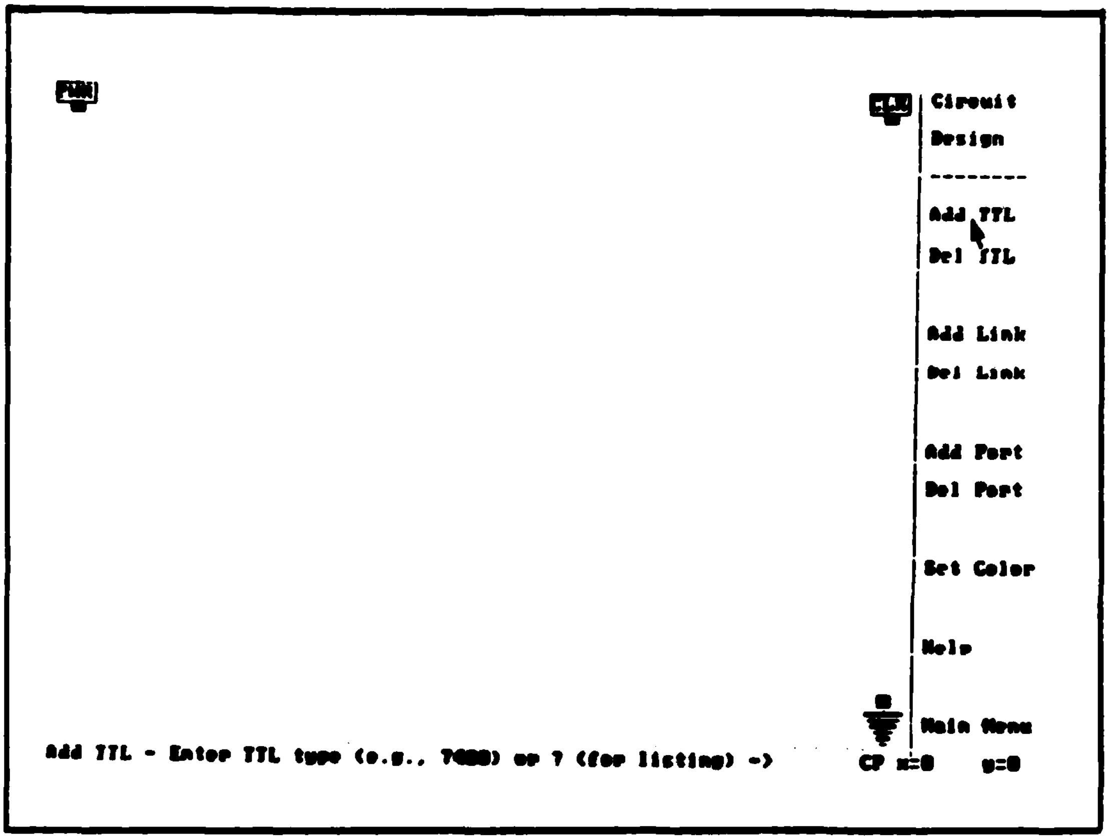


Figure A-5. Clicking on "Design Circuit" of "Main Menu"

The "Design Circuit" option title will become highlighted and remain this way until the menu switches. The first time the "Design Circuit" option is selected and a circuit has not been retrieved, the initial file configurations (i.e., placing the power, ground and clock icons in the window and creating the necessary temporary files) for a circuit are established. The user is advised on the communication line: "Preparing system for circuit design - please be patient". Once the preparation is completed, the "Circuit Design" menu is displayed. Once the "Circuit Menu" is displayed, changes to the current circuit design can be made using the "Add TTL", "Del TTL", "Add Link", "Del Link", "Add Port", "Del Port", or "Set Color" options described below. (See Figure A-2).

NOTE: DUE TO THE PROGRAMMING, ONLY PART OF THE "DESIGN CIRCUIT" TITLE IS HIGHLIGHTED AT ANY ONE TIME. THE PART HIGHLIGHTED IS BASED ON WHICH WORD THE MOUSE CLICKED ON. IT MAKES NO DIFFERENCE WHICH PART OF THE TITLE IS SELECTED

Adding a TTL. To add a TTL to the circuit the user clicks on the "Add TTL" option of the "Circuit Design" menu. The user is then prompted on the communication line with: "Add TTL - Enter TTL type (e.g., 7400) or ? (for listing) -> ". (See Figure A-6).

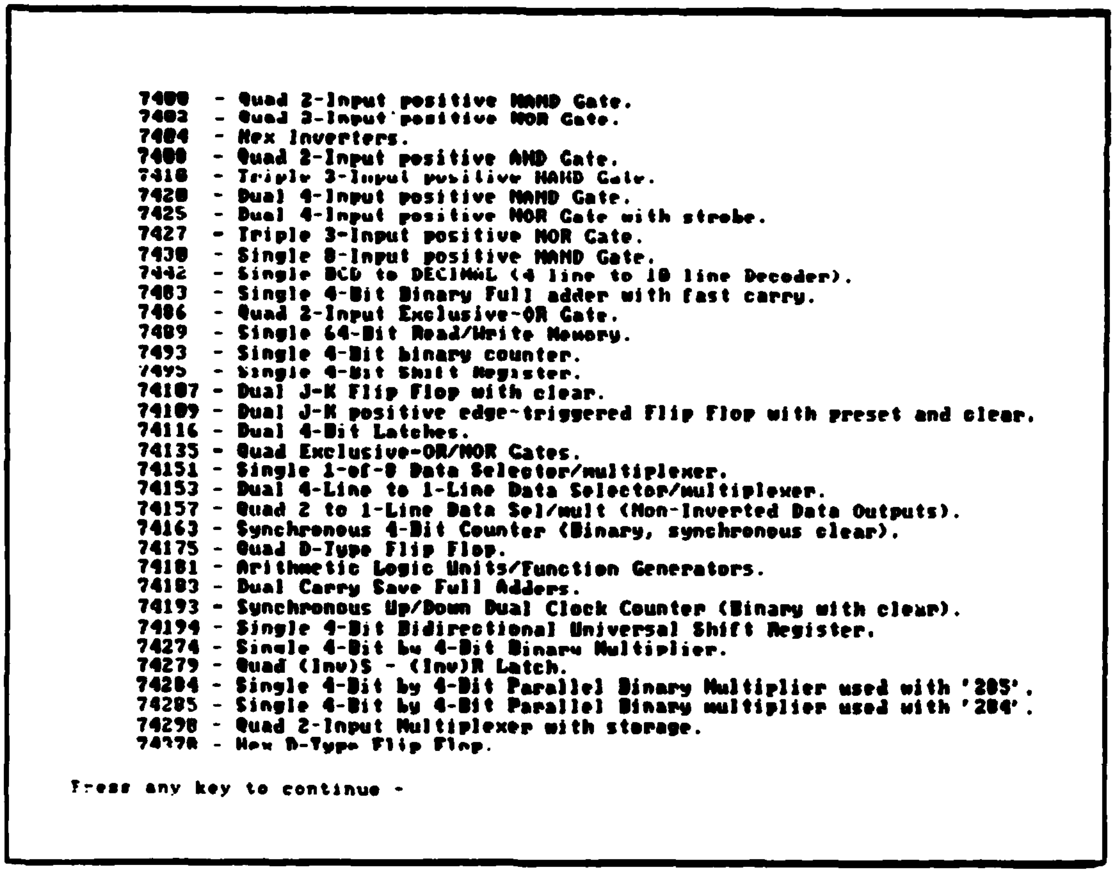


Add TTL - Enter TTL type (e.g., 7400) or ? (for listing) ->

CP x=0 y=0

Figure A-6. "Add TTL" option of "Circuit Design" menu

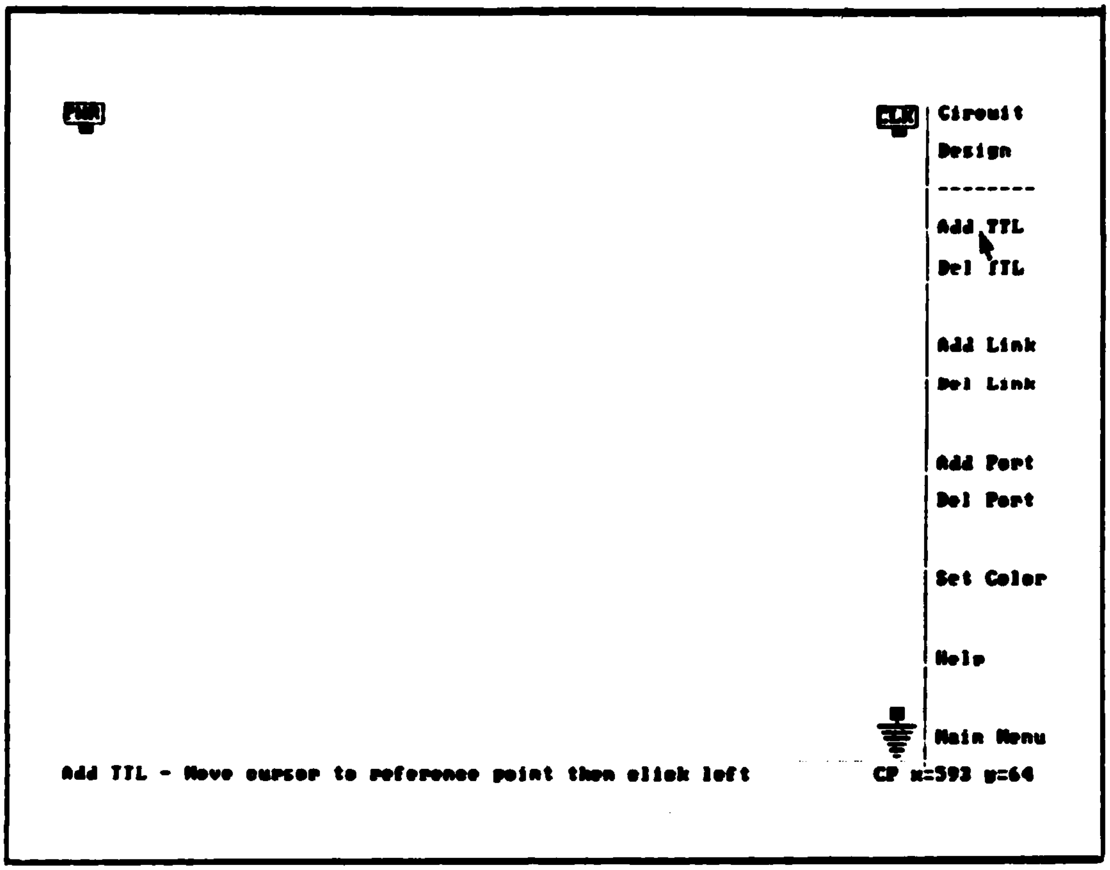
At this point the user can either enter the TTL type, if known, or enter "?" for a listing of the TTLs that are available for use. If the user enters the "?", the current screen is replaced by a screen identifying the TTL number and its description. (See Figure A-7).



```
7400  - Quad 2-Input positive NAND Gate.
7402  - Quad 2-Input positive NOR Gate.
7404  - Hex Inverters.
7408  - Quad 2-Input positive AND Gate.
7410  - Triple 3-Input positive NAND Gate.
7420  - Dual 4-Input positive NAND Gate.
7425  - Dual 4-Input positive NOR Gate with strobe.
7427  - Triple 3-Input positive NOR Gate.
7430  - Single 8-Input positive NAND Gate.
7442  - Single BCD to DECIMAL (4 line to 10 line Decoder).
7483  - Single 4-Bit Binary Full adder with fast carry.
7486  - Quad 2-Input Exclusive-OR Gate.
7489  - Single 64-Bit Read/Write Memory.
7493  - Single 4-Bit binary counter.
7495  - Single 4-Bit Shift Register.
74107 - Dual J-K Flip Flop with clear.
74109 - Dual J-K positive edge-triggered Flip Flop with preset and clear.
74116 - Dual 4-Bit Latches.
74135 - Quad Exclusive-OR/NOR Gates.
74151 - Single 1-of-8 Data Selector/multiplexer.
74153 - Dual 4-Line to 1-Line Data Selector/multiplexer.
74157 - Quad 2 to 1-Line Data Sel/mult (Non-Inverted Data Outputs).
74163 - Synchronous 4-Bit Counter (Binary, synchronous clear).
74175 - Quad D-Type Flip Flop.
74181 - Arithmetic Logic Units/Function Generators.
74183 - Dual Carry Save Full Adders.
74193 - Synchronous Up/Down Dual Clock Counter (Binary with clear).
74194 - Single 4-Bit Bidirectional Universal Shift Register.
74274 - Single 4-Bit by 4-Bit Binary Multiplier.
74279 - Quad (Inv)S - (Inv)R Latch.
74284 - Single 4-Bit by 4-Bit Parallel Binary Multiplier used with '285'.
74285 - Single 4-Bit by 4-Bit Parallel Binary multiplier used with '284'.
74298 - Quad 2-Input Multiplexer with storage.
74378 - Hex D-Type Flip Flop.

Press any key to continue -
```

Figure A-7. Available types of TTL chips

The user must press any key to return to the menu. Once the menu has returned, the user is prompted with: "Enter TTL type (Do not use spaces) ->", and the "?" option is removed until the next time the user clicks on "Add TTL". If the user enters a valid TTL type, the following advisory is displayed: "Add TTL - Move cursor to reference point then click left". (See Figure A-8).



```
Add TTL - Move cursor to reference point then click left          CP x=393 y=64
```

Figure A-8. Positioning a TTL chip

Once the user points the cursor at a valid location (place where the TTL will not overlap other TTLs, ports, power icon, clock icon, ground icon or the design window edge) within the design window and clicks the left mouse button, a TTL will be positioned with its top edge at the specified Y-coordinates and with the left edge of pins at approximately the location specified (see note). The TTL will be displayed with its type (i.e., 7400) and a number that provides a unique id. (See Figure A-9).

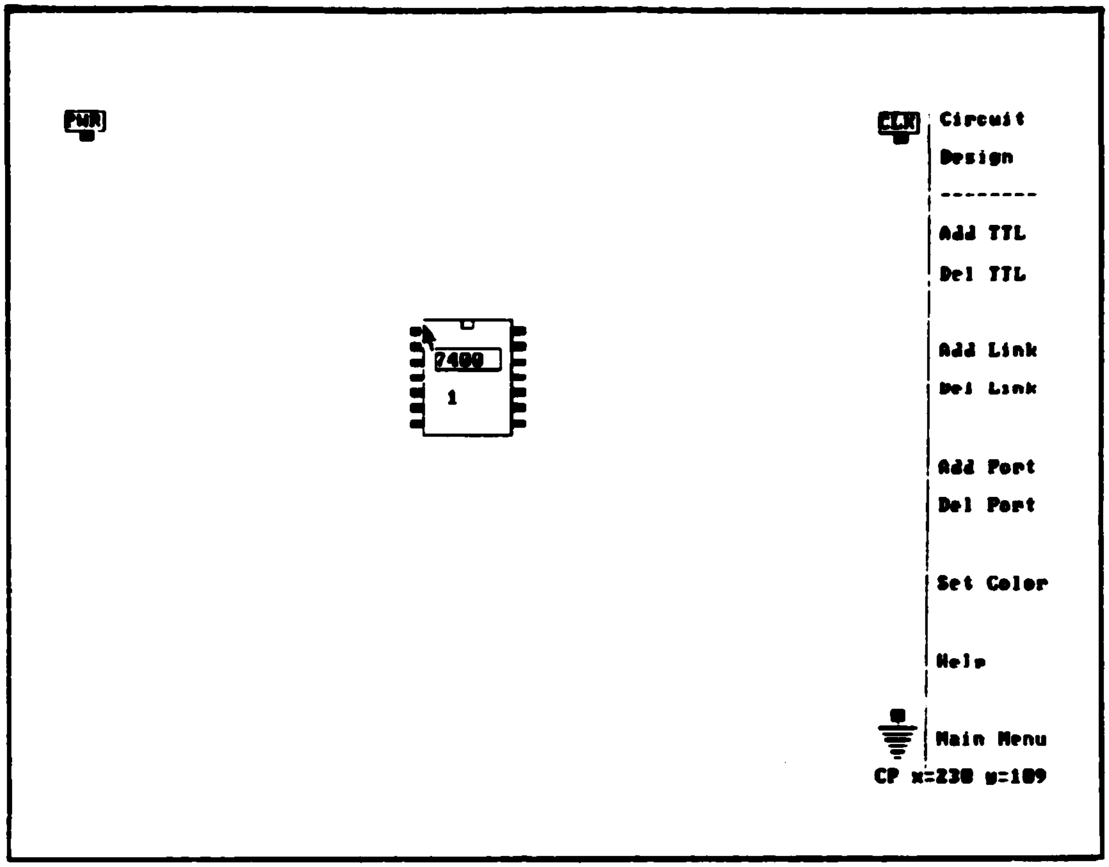


Figure A-9. Displayed TTL

NOTE: THE ACTUAL POSITION OF THE TTL IS DEPENDENT UPON THE X-COORDINATE. ALL TTL REFERENCE POINTS ARE BASED ON THE X-COORDINATE BEING MULTIPLES OF 8 STARTING WITH 8. THIS IS TO ALLOW FOR THE TEXT DISPLAYED ON THE TTL. SPACING AROUND ALL BUT THE TOP IS PROVIDED. IF A TTL IS DISPLAYED WITH AN ID GREATER THAN 319 THE USER SHOULD REDESIGN THE CIRCUIT FROM SCRATCH BECAUSE LOGSIM WILL BE UNABLE TO UNIQUELY IDENTIFY THIS TTL. IF THE USER DECIDES NOT TO ADD A TTL, EITHER SPECIFY AN INVALID TTL OR USE THE "Del TTL" OPTION.

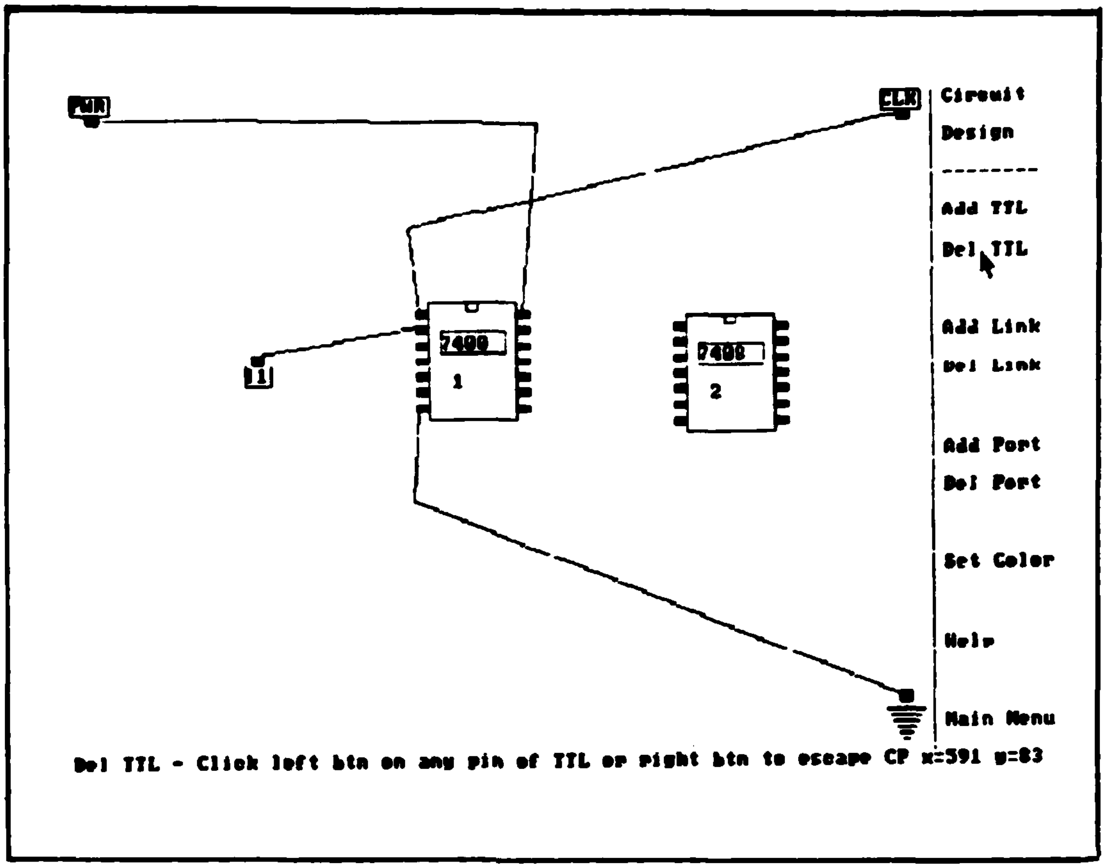
Error conditions:

If the user types in more than five characters , the system advises: "Sorry - Do not use leading/trailing spaces - try again ".

If the number of characters is less than six and an invalid TTL type or any blank spaces are inserted, the user is advised: "The specified TTL is not available -- press any key to continue". Once the user presses any key, the "Add TTL" title no longer is highlighted, no TTL is added, and the user is returned to the status pricr to selecting this menu option.

If the user attempts to place a TTL at a position where it will not remain within the design window or if it may overlap other TTLs, ports, or icons the user is advised: "Invalid coordinates -- press any key to continue". Once the user presses any key the add option is terminated and no TTL added.

Deleting a TTL. All links must first be removed from a TTL before it can be deleted. To delete a TTL, the user clicks on the "Del TTL" option of the "Circuit Design" menu. The option title will become highlighted and the user will be prompted with: "Del TTL - Click left btn on any pin of TTL or right btn to escape". (See Figure A-10).



Del TTL - Click left btn on any pin of TTL or right btn to escape CP x=591 y=83

Figure A-10. Deleting a TTL

If the user presses the right button, the delete option is terminated and no action taken. However, if the user clicks the left button while pointing at a pin of a TTL (and the TTL has no links attached to any of its pins), the advisory: "Deleting this TTL takes time - please be patient" is presented. When the deletion is completed the TTL is removed, the advisory is also removed, and the "Del TTL" title is no longer highlighted.

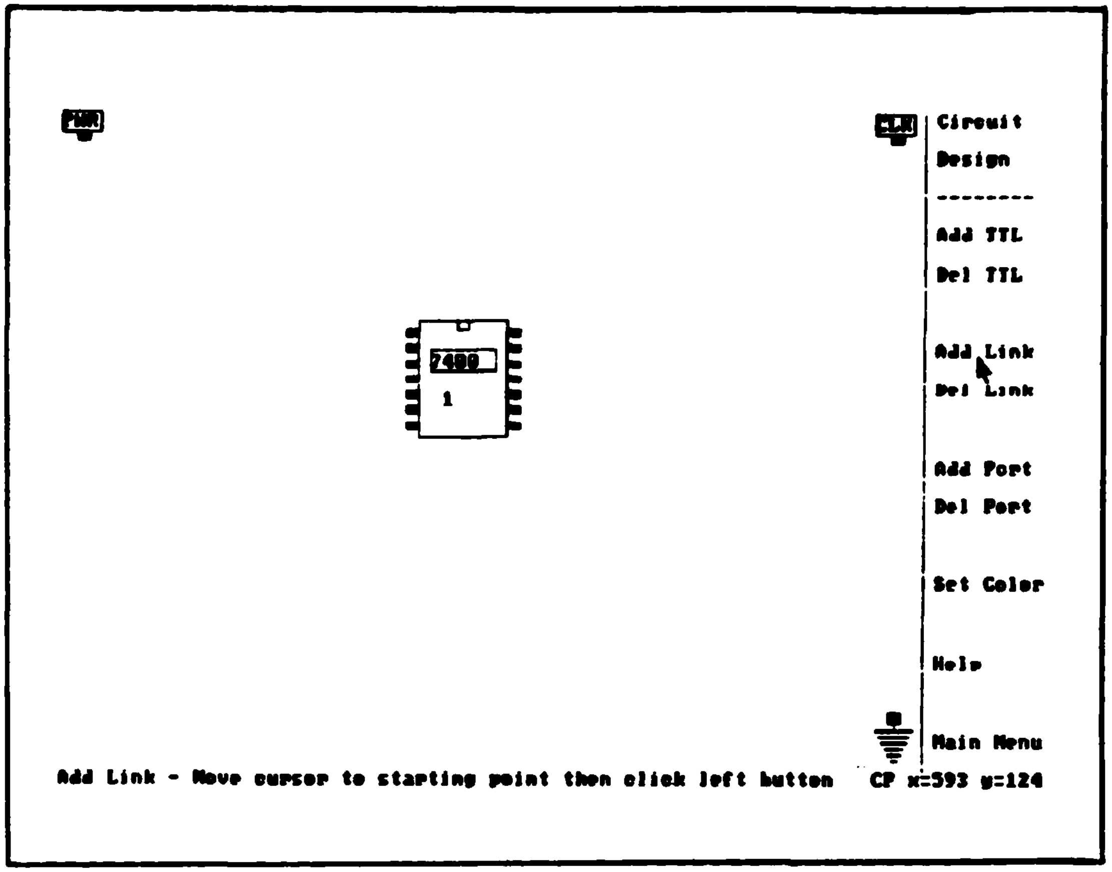
Error conditions:

If the user clicks on a pin of a TTL and that TTL has at least one link connected to it, the following advisory is given: "All links to this TTL must be removed - press any key to cont-". Once the user presses any key the delete option is also terminated and no action taken.

If the user clicks on a pin connection of something other than a TTL, the advisory: "Sorry this was not identified as a TTL - press any key to cont-". Once the user presses any key the delete option is terminated and no action taken.

If the user does not click on any pin connection, the following advisory is displayed: "Invalid coordinates -- press any key to continue". Once the user presses any key the delete option is terminated and no action taken.

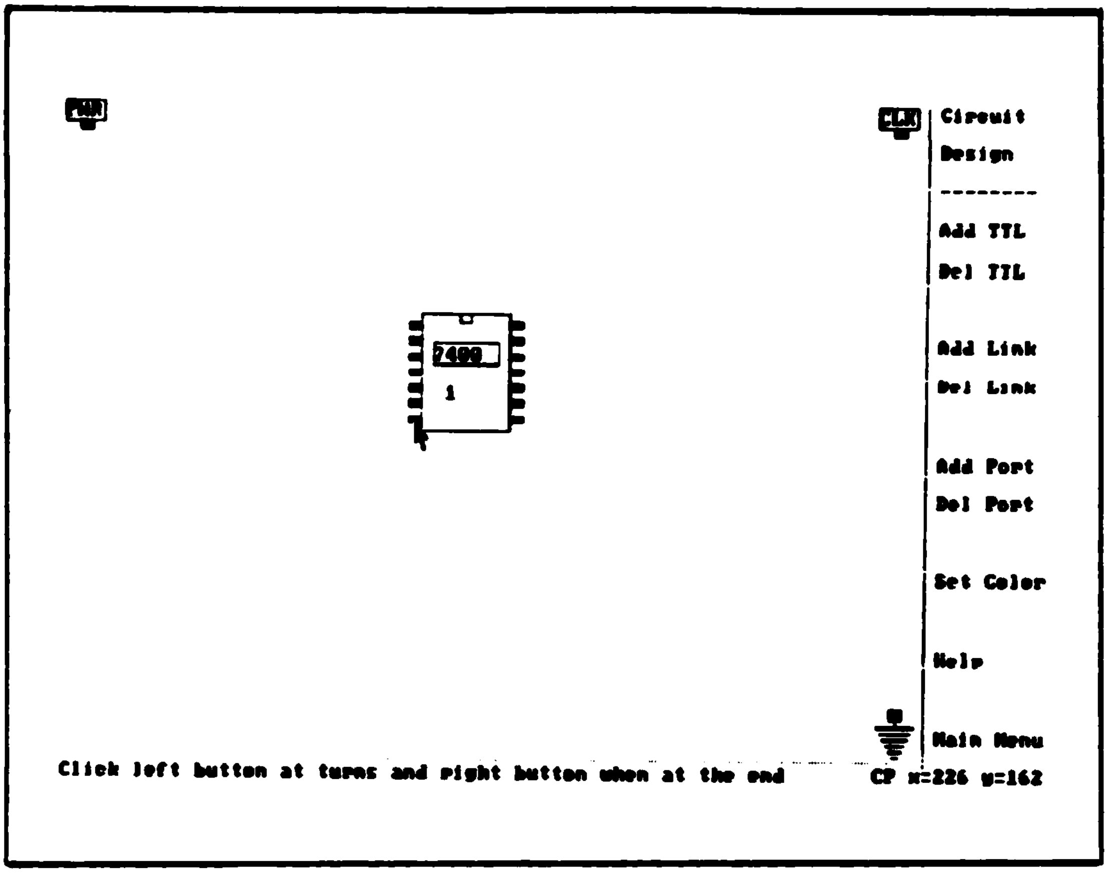
Adding a link. To add a link the user clicks on the "Add Link" option of the "Circuit Design" menu. The option title will remain highlighted until the option is terminated. (See Figure A-11).



Add Link - Move cursor to starting point then click left button   CP x=593 y=124

Figure A-11. Clicking on "Add LINK" option

The user may quit this option by clicking the right button prior to identifying the links starting point. The user must start and terminate all links at pin connections (represented on the screen by a shaded box). The starting point and all turning points are specified by clicking at the appropriate point using the left button. (See Figure A-12).



Click left button at turns and right button when at the end

CP x=226 y=162

Figure A-12. Pointing to link starting point

The ending point must be terminated using the right button. If the user clicks on a valid starting point, the user is allowed up to 10 turning points and an advisory: "Click left button at turns and right button at the end". To terminate the link, the user clicks right button on the valid ending point and the "Add Link" title is returned to normal color. (See Figure A-13).

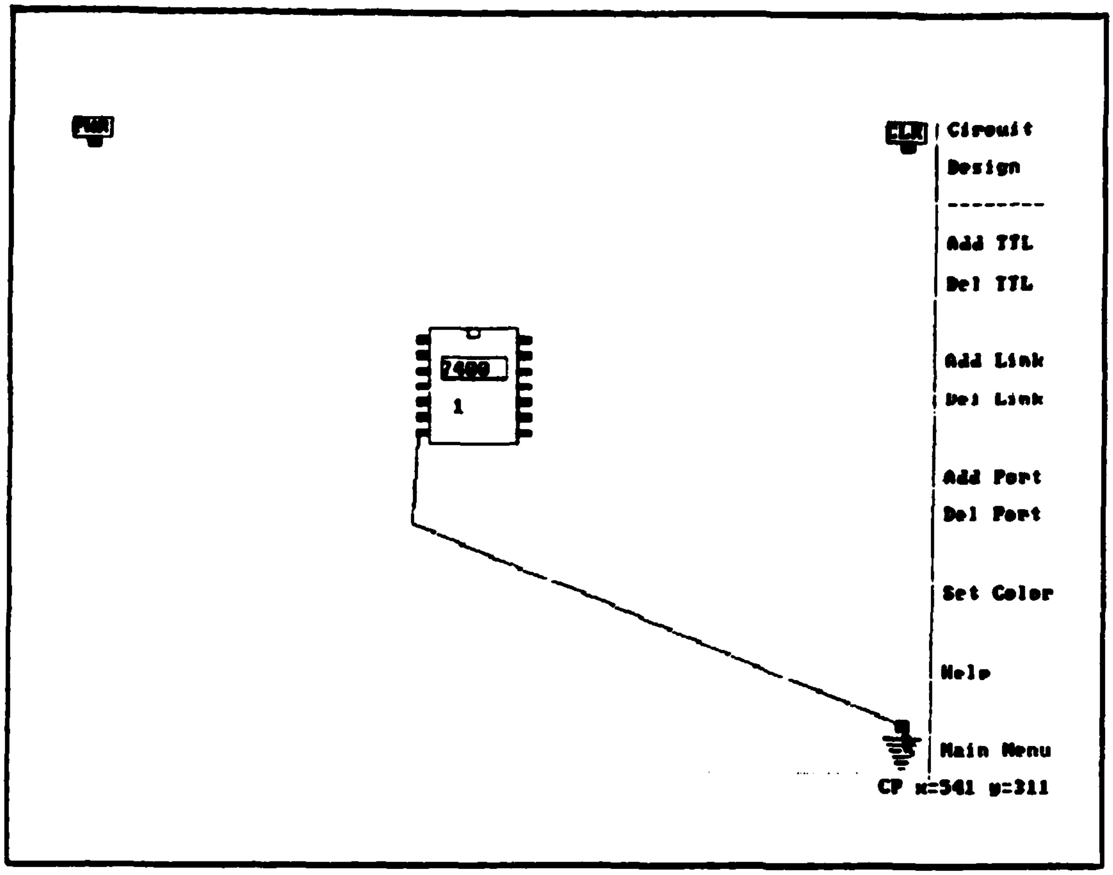


Figure A-13. Pointing to link ending point

If the user clicks the right button at an invalid point or if the user exceeds 10 turning points, the line is not drawn to the specified point. The user can not terminate the "Add Link" option until a valid pin location is specified.

NOTE: INTERMEDIATE POINTS IN A LINK DO NOT ESTABLISH CONNECTIONS (i.e., ONLY THE END POINTS OF A LINK ARE CONNECTED). IN ADDITION, CONNECTING PIN A TO PIN B AND THEN CONNECTING PIN B TO PIN C, DOES NOT RESULT IN PIN A BEING CONNECTED TO PIN C. THE USER MUST EXPLICITLY MAKE ALL CONNECTIONS. FOR EXAMPLE, TO ESTABLISH THE POWER CONNECTION TO THREE TTLs, THE USER MUST ESTABLISH THREE LINKS, EACH STARTING AT THE PIN FOR THE POWER AND ENDING AT THE APPROPRIATE PIN (i.e., PIN 14 ON A 7400) ON EACH TTL. FURTHERMORE, THE USER SHOULD USE THE CURRENT POSITION (CP) INDICATOR IN THE LOWER RIGHT CORNER WHEN DRAWING LINKS. DRAWING LINKS WITH MINOR X AND Y COORDINATE CHANGES HAVE A

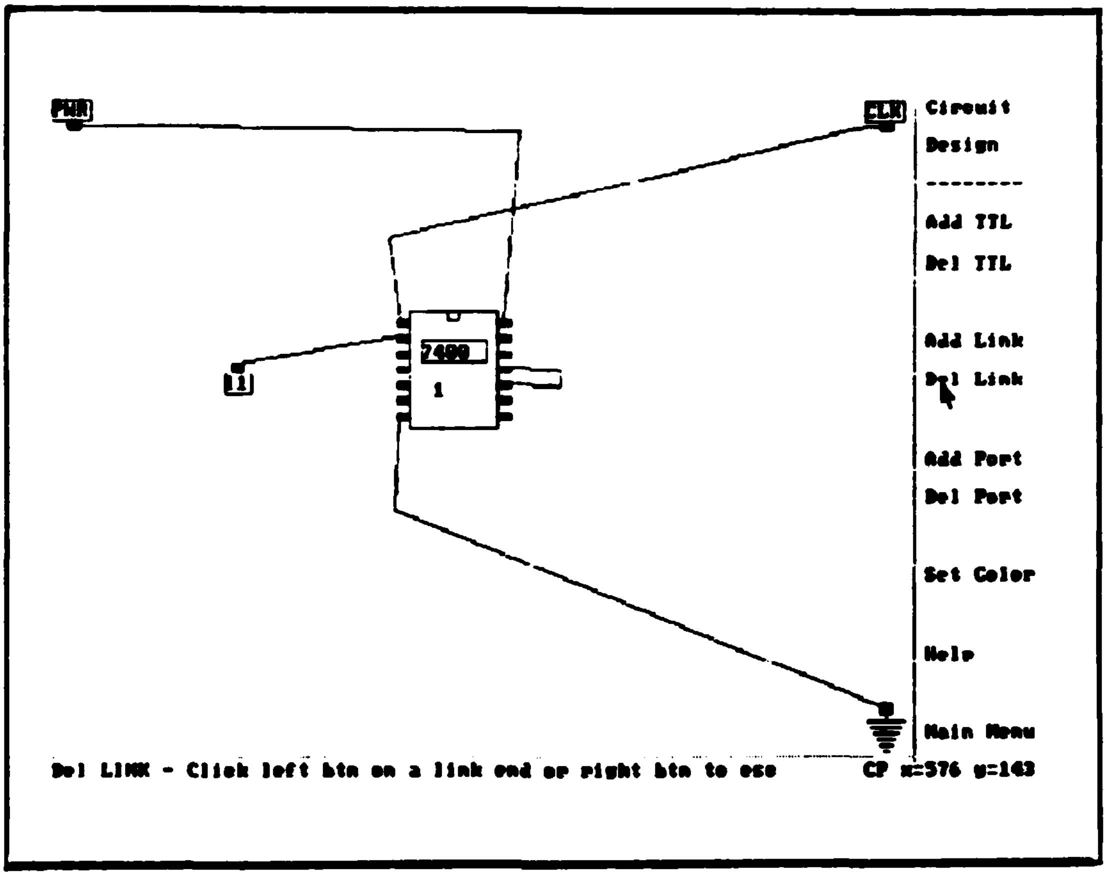
TENDENCY TO MAKE THE LINKS LOOK BROKEN, ALTHOUGH THEY ARE NOT.

Error conditions:

If the user attempts to start a link anywhere but at a pin connection the system will advise: "Invalid coordinates -- press any key to continue". Once the user presses a key the add link option is terminated.

If the user attempts to put a turning point or end the link outside of the design window, the advisory "Invalid coordinates -- press any key to continue" message will be presented.

Deleting a link. To delete a link, the user clicks on the "Del Link" option in the "Circuit Design" menu. The option title will become highlighted until terminated. The user may quit this option by depressing the right button. The user will be prompted with: "Click left btn on a link end or right btn to esc". (See Figure A-14).



Del LINK - Click left btn on a link end or right btn to esc   CP x=576 y=143

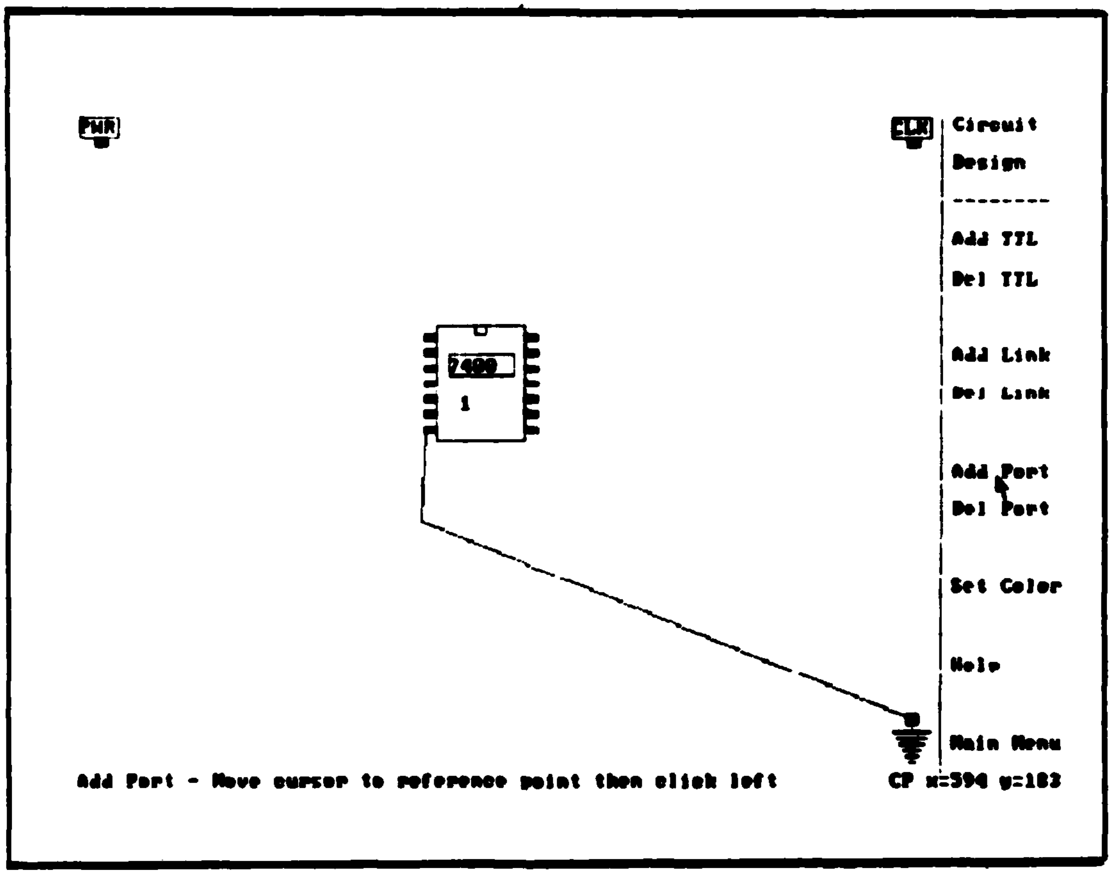
Figure A-14. Clicking on "Del Link" option

If the user selects a valid point, a second prompt will be presented: "Now click left button on the other end of link". The program will display messages about testing for the link and deleting the link if applicable or it will advise that the link does not exist.

Error conditions:

If the user does not click on a valid pin location, an "Invalid coordinates -- press any key to continue" message will be presented.

Adding an input port. To add an input port to the circuit the user clicks on the "Add Port" option of the "Circuit Design" menu. The user is then prompted with: "Add Port - Move cursor to reference point then click left". (See Figure A-15).



Add Port - Move cursor to reference point then click left

CP x=394 y=183

Figure A-15. Clicking on "Add Port" option

Once the user points the cursor at a valid location (place where the port will not overlap other TTLs, ports, power icon, clock icon, ground icon or the design window edge) within the design window and clicks the left mouse button, the user is prompted with: "Add Port - Enter a two character name for this port ->". The two characters can include blank spaces. Once two characters are entered a port will be positioned with its top at the specified Y-coordinates and with the left edge of the port at approximately the location specified (see note). The port will be displayed with its label (e.g., I1). (See Figure A-16).

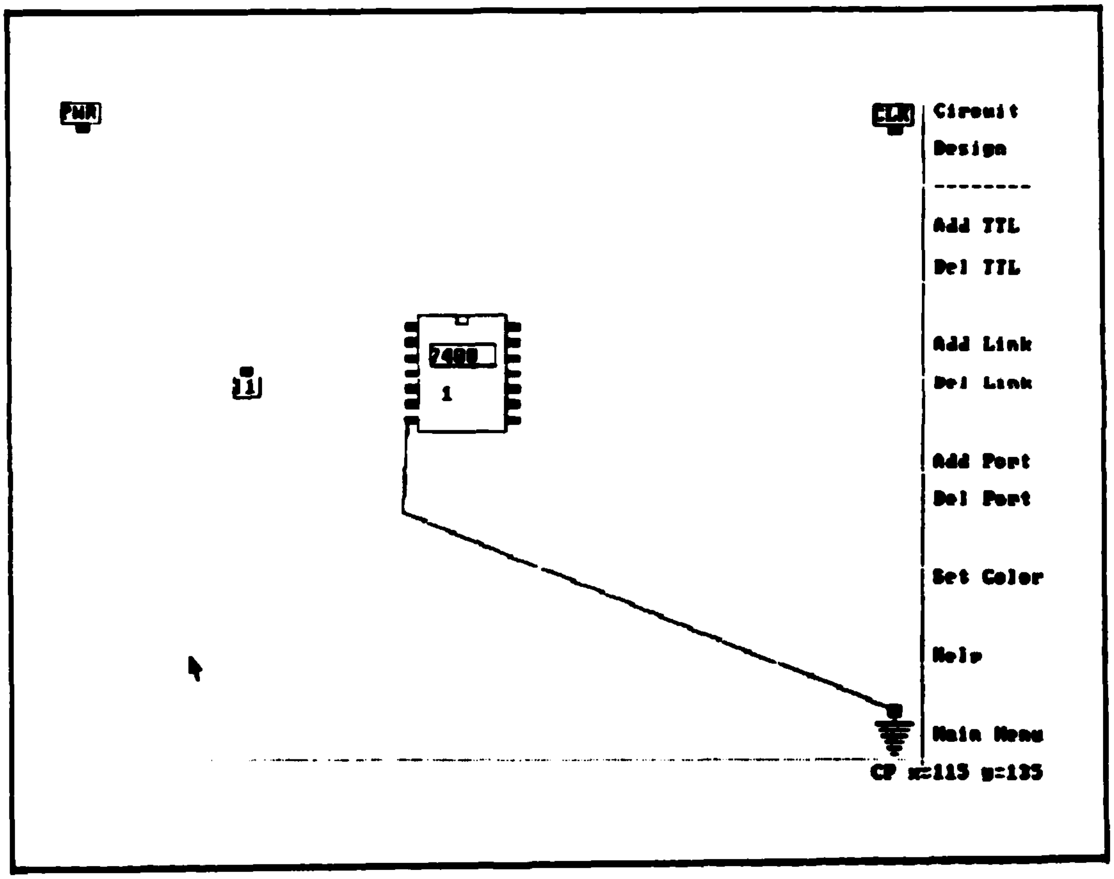


Figure A-16. Display of input port

NOTE: THE ACTUAL POSITION OF THE PORT IS DEPENDENT UPON THE X-COORDINATE. ALL PORT REFERENCE POINTS ARE BASED ON THE X-COORDINATE BEING MULTIPLES OF 8 STARTING WITH 8. THIS IS TO ALLOW FOR THE TEXT DISPLAYED ON THE PORT. SPACING AROUND ALL BUT THE TOP IS PROVIDED. THE TWO CHARACTER NAME CAN BE

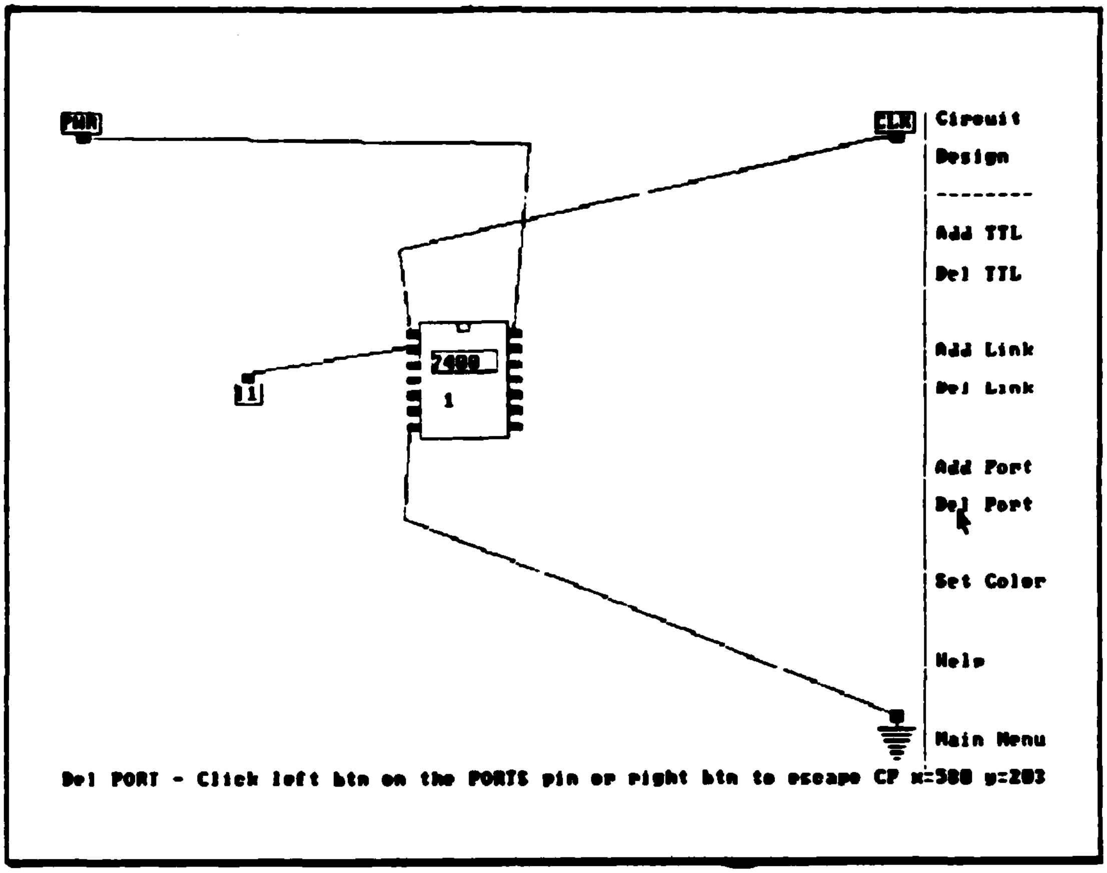
UPPER OR LOWER CASE AND STILL BE UNIQUE; HOWEVER, THE USER SHOULD NOT USE THE SAME EXACT TWO CHARACTER NAME MORE THAN ONCE, OTHERWISE CONFUSION WILL OCCUR WHEN USING THE INPUT OPTION FOR LOGSIM. IF THE USER DECIDES NOT TO ADD AN INPUT PORT, EITHER CLICK THE RIGHT BUTTON PRIOR TO SPECIFY A LOCATION IN THE DESIGN WINDOW OR USE THE "Del Port" OPTION.

Error conditions:

If the user positions the pointer at a place where the port will not remain within the design window or if it may overlap other TTLs or ports, the user is advised: "Invalid coordinates -- press any key to continue". Once the user presses any key the add option is terminated and no port added.

If the user inputs a title for the port with any amount other than two characters, the following prompt is given: "SORRY - Must be two character name - try again ->".

Deleting an input port. All links must be removed from the port before it can be deleted. To delete an input port, the user clicks on the "Del Port" option of the "Circuit Design" menu. The option title will become highlighted and the user will be prompted with: "Del PORT - Click left btn on the PORTS pin or right btn to escape". (See Figure A-17).



```
Del PORT - Click left btn on the PORTS pin or right btn to escape CP x=380 y=203
```

Figure A-17. Clicking on "Del Port" option

If the user presses the right button, the delete option is terminated and no action taken. However, if the user clicks on the pin of an input port with no links to it, the advisory: "Deleting this PORT takes time - please be patient". When the deletion is completed the port is removed, the advisory is also removed, and the "Del Port" title is no longer highlighted.

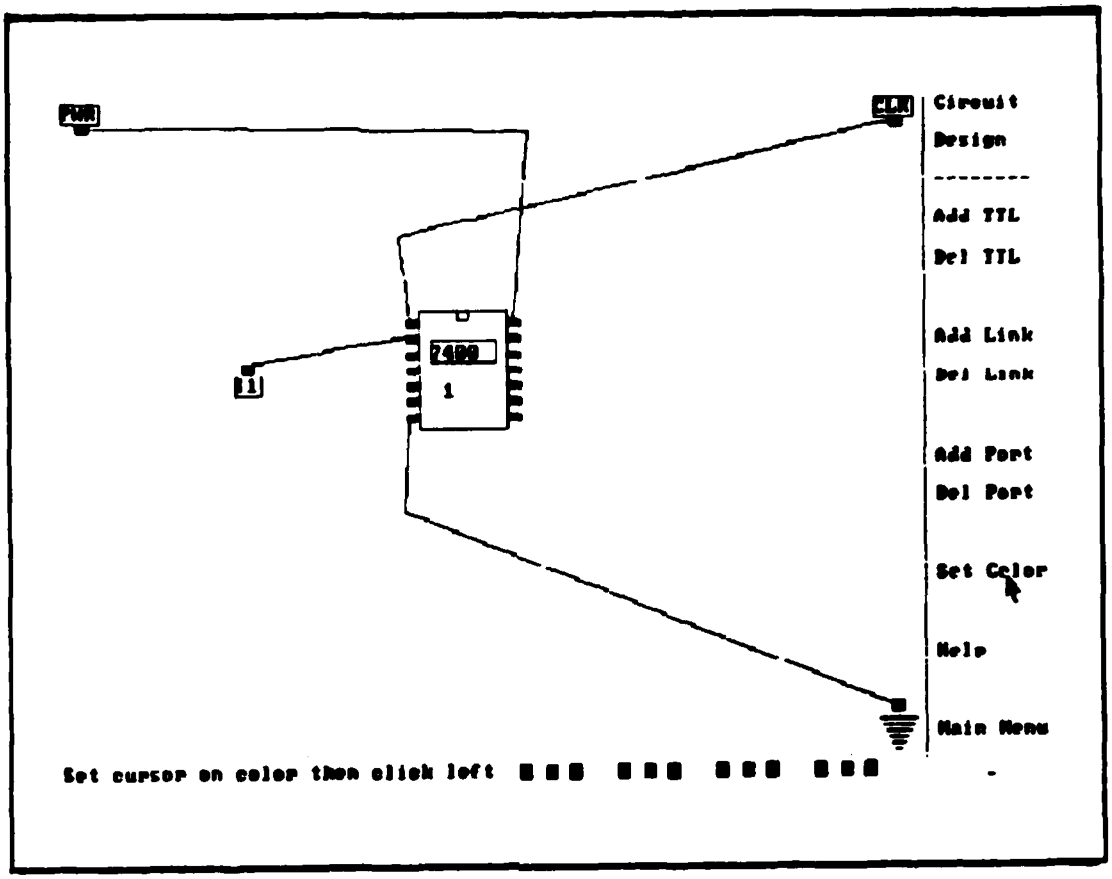
<u>Error conditions</u>:

If the user clicks on a pin connection of something other than an input port, the advisory: "Sorry this was not identified as a PORT - press any key to cont-". Once the user presses any key the delete option is terminated and no action taken.

If the user clicks on anything but a pin connection, the following advisory is displayed: "Invalid coordinates -- press any key to continue". Once the user presses any key the delete option is terminated and no action taken.

If the user clicks on the pin of a port with at least one link connected to it, the following advisory is given: "All links to this PORT must be removed - press any key to cont-". Once the user presses any key the delete option is also terminated and no action taken.

Setting the color. To select a new foreground color the user must click on the "Set color" option of the "Circuit Design" menu. The option title will become highlighted and remain highlighted until either a color is selected or option terminated. Once the title is highlighted the user will be prompted with: "Set cursor on color then click left". (See Figure A-18).



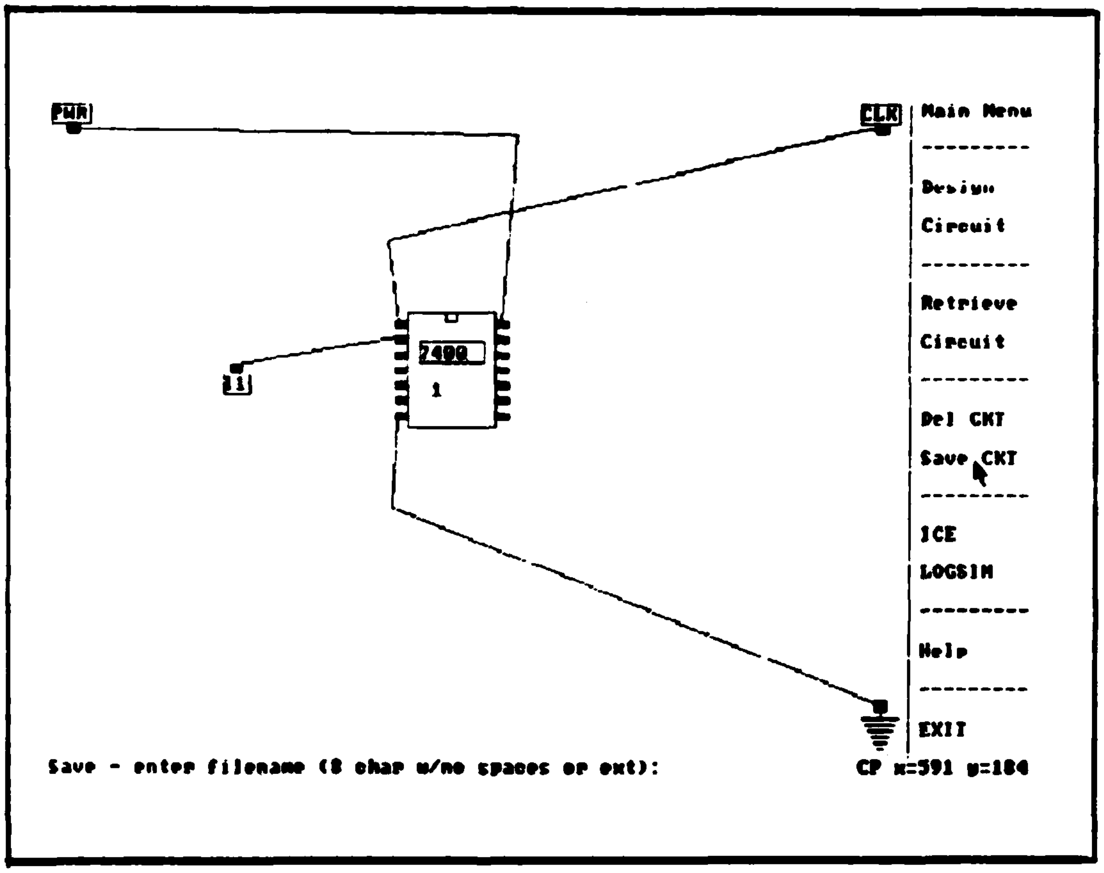
Set cursor on color then click left ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■

Figure A-18. Clicking on "Set Color" option

The user can click the left mouse button while pointing to any color selection (i.e., the colored boxes on the communications line) and that will become the color for future designs until changed. The "Set color option is then terminated. If the user clicks the right button or points to the background color, no color change is made and the option is terminated.

III. Saving and Retrieving a circuit

Saving a circuit. To save a circuit the user clicks on the "Save CKT" option in the "Main Menu". The "Save CKT" option title will become highlighted. (See Figure A-19).



```
Save - enter filename (8 char w/no spaces or ext):          CP x=591 y=184
```

Figure A-19. Clicking on the "Save" option

If a circuit is present, the user is prompted with "Save - enter filename (8 char w/no spaces or ext): ". The user can enter one to eight valid characters for the filename and press "RETURN"; however, the user should not enter any blank spaces or extensions, because the necessary files will be saved using the filename provided with predetermined extensions (i.e., .icn, .loc, .grf, .ckt, .in, .ind, .dis, .iot, wav, and .out). Entering blanks or extensions could result in files not being saved. Once the user has entered one to eight characters and the "RETURN", the user will be advised, "Saving this circuit takes time - please be patient". This indicates that a copy of the current design and all related files are being saved to the filename specified. Once the files have been saved the advisory is removed, the "Save CKT" title is returned to normal, and the user is returned to the screen display prior to selecting this option. If the user inserts the filename of an existing circuit design, the user will be advised: "This file already exists - Do you wish to overwrite it <y or n>". If the user responds with a negative answer, the system prompts for another filename.

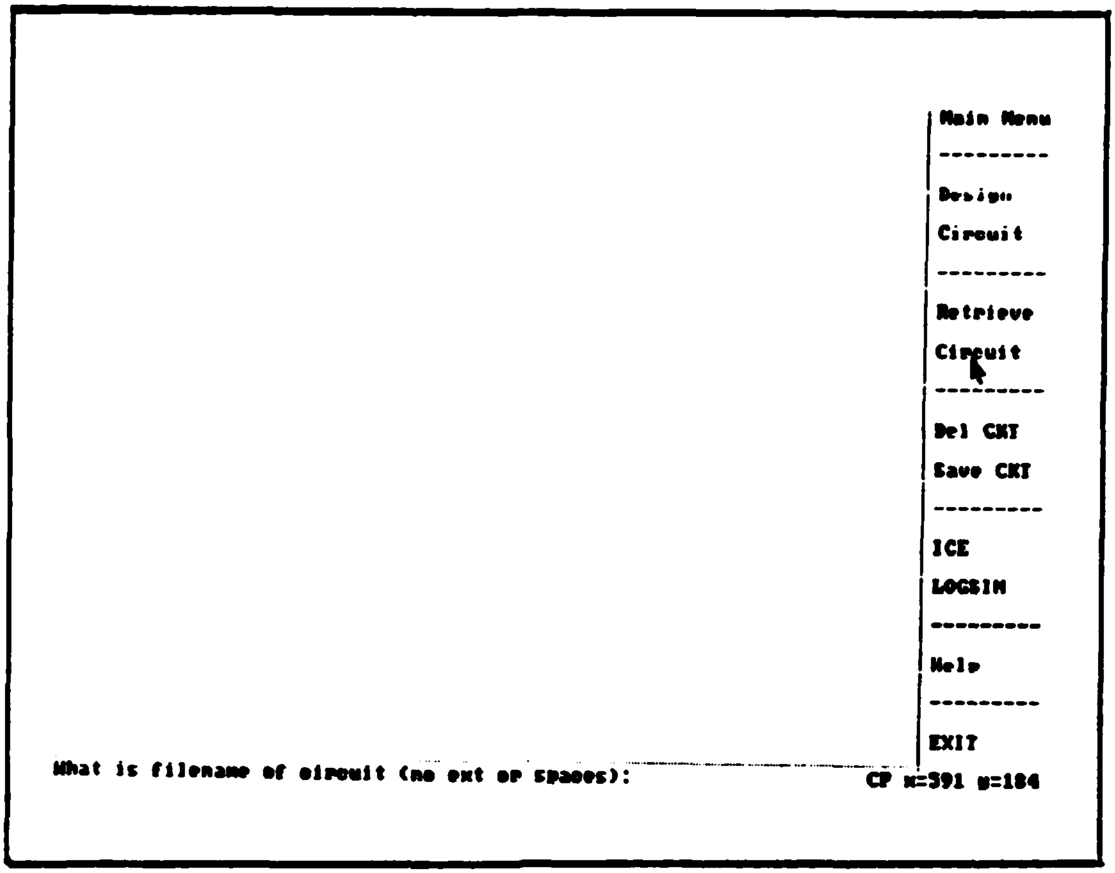
CAUTION: THE USER SHOULD NOT USE THE FILENAME "TEMP" BECAUSE ALL TEMP FILES ARE DELETED WHEN THE PROGRAM IS TERMINATED.

Error conditions:

If the user has not yet created a circuit design (must have at least the power, ground, and clock icons displayed in the circuit design window) or retrieved an existing circuit, a prompt will be presented stating "You have not created/retrieved a circuit - press any key to cont-". The user then presses any key and the "Save CKT" title is returned to normal, the prompt is removed, and the user is returned to the previous screen display.

If the user inserts an incorrect number of characters for the filename, the following prompt will be displayed: "8 characters only - try again -> ". Once the user has entered one to eight characters and the "RETURN", the program will continue with the next step.

Retrieving a circuit. To retrieve a circuit the user clicks on the "Retrieve Design" option in the "Main Menu". Whichever word the user clicked on will remain highlighted while this option is active. Prior to entering this option the user must know the filename of the design. Currently the only ways to know the filenames are to have saved the design previously and remember its name or to return to the host computer's operating system and review the filenames in the same directory as this program (graph.exe). Files that have a filename containing the extensions ".loc" and ".icn" are possible candidates since circuits that are saved have these filename extensions. However, other software programs may use files with these same extensions, but are not used by this program. If there is no current circuit design in progress, the user is prompted with "What is filename of circuit (no ext or spaces):". (See Figure A-20).



What is filename of circuit (no ext or spaces):

CP x=391 y=184

Figure A-20. Valid "Retrieve Circuit" request

Enter the filename with no extensions, followed by a "RETURN". If the filename is valid, the user is advised on the communications line: "Retrieving this circuit takes time - please be patient". When the circuit has been retrieved, the circuit is redrawn in the window, the advisory is removed, and the retrieve circuit option is no longer highlighted. Once the circuit has been retrieved, changes are made to the circuit using the "Design Circuit" option in the "Main Menu".

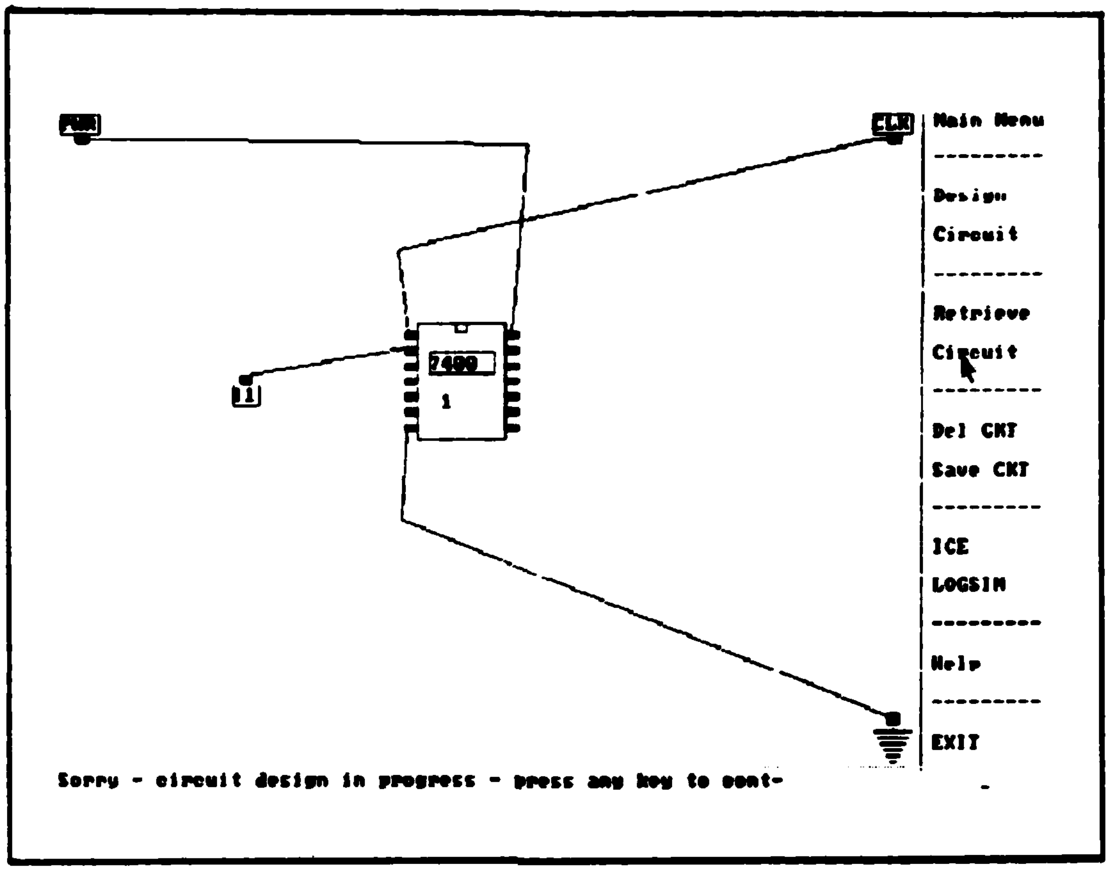
NOTE: DUE TO THE PROGRAMMING, ONLY PART OF THE "RETRIEVE DESIGN" TITLE IS HIGHLIGHTED AT ANY ONE TIME. THE PART

HIGHLIGHTED IS BASED ON WHICH WORD THE MOUSE CLICKED ON. IT MAKES NO DIFFERENCE WHICH PART OF THE TITLE IS SELECTED

Error conditions:

If the filename entered is not valid then the user is advised: "This filename does not exist - press any key to cont-". The user then presses any key and the advisory is removed, the appropriate option is no longer highlighted, and the user is at the point prior to this menu selection.

If a circuit is currently in the circuit design window (at least the power, ground, and clock icons displayed), the user will be advised: "Sorry - circuit design in progress - press any key to cont- ". (See Figure A-21).



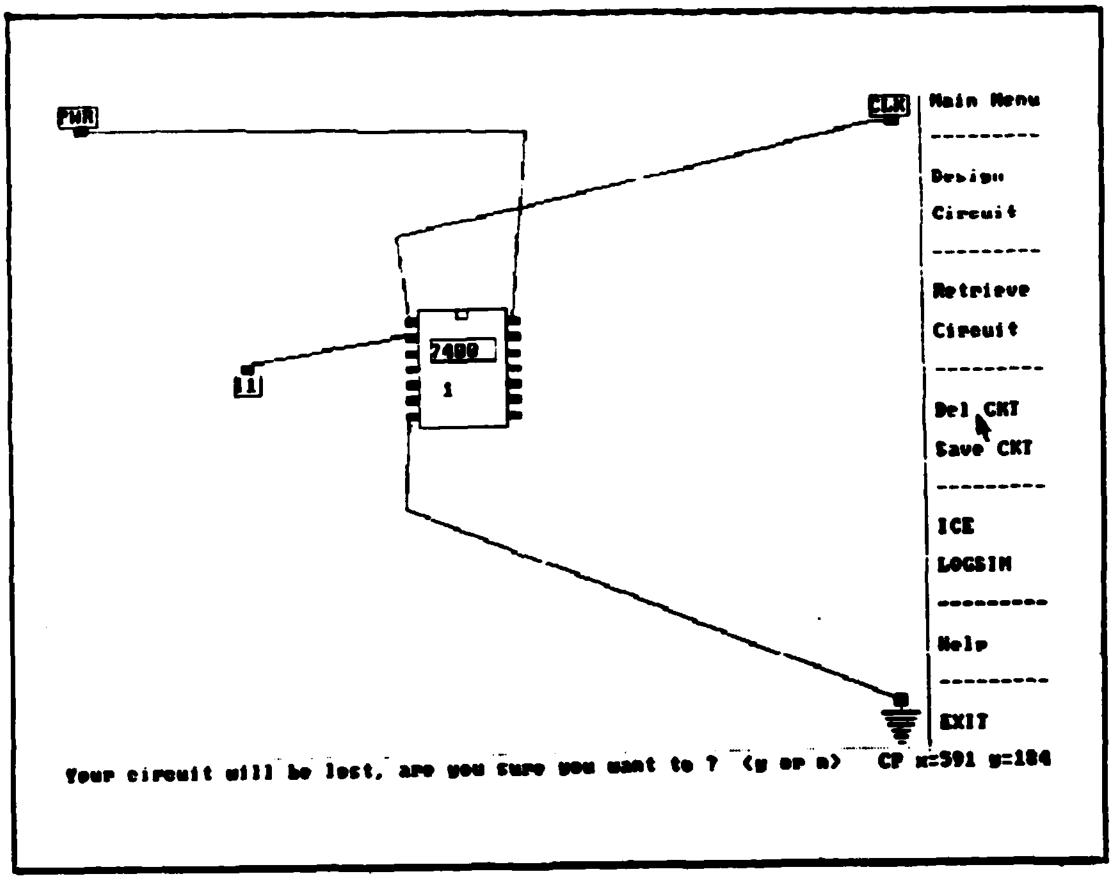
Sorry - circuit design in progress - press any key to cont-

Figure A-21. Invalid "Retrieve Circuit" request

Pressing any key returns the user to the design in progress. The only way to retrieve a circuit after a design is in progress is to first delete the current design, using the "Del CKT" option in the "Main Menu" or exiting the program and restarting. The user must save the current design circuit prior to using either of these options, if this circuit is to be retrieved later.

IV. Deleting a circuit:

To delete the circuit in the design window, the user must click on the "Del CKT" option in the "Main Menu". Once this option has been selected the "Del CKT" will be highlighted and the user will be prompted with "Your circuit will be lost, are you sure you want to ? <y or n> ". (See Figure A-22).



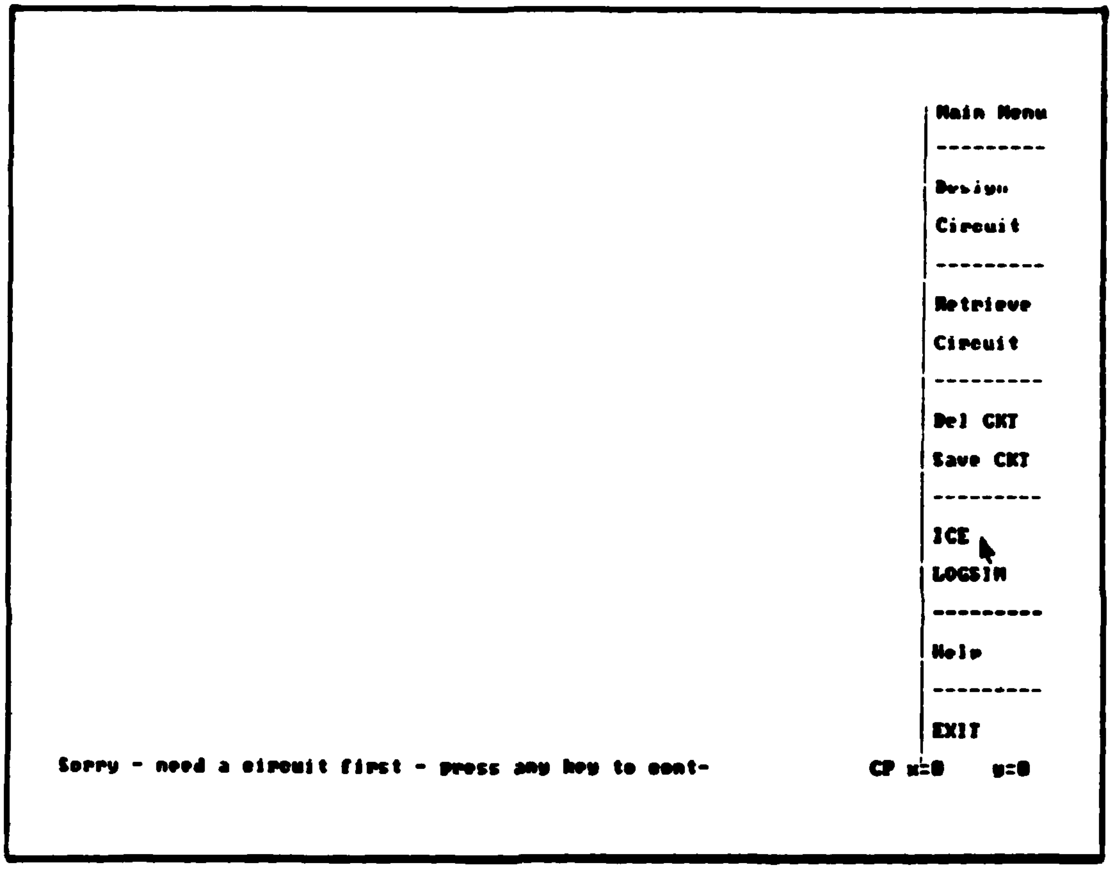
```
Your circuit will be lost, are you sure you want to ?  <y or n>    CP x=591 y=184
```

Figure A-22. Clicking on "Del CKT" option

Entering a "Y" or "y" results in the circuit design window being cleared, the circuit being lost (if not saved prior to this option), the "Del CKT" option being no longer highlighted, and the user returning to the condition as when starting this program. If the user presses any other key the "Del CKT" option is ignored, "Del CKT" is no longer highlighted, the prompt is removed and the user is returned to the previous status.

V. Using the ICE tool:

To use any of the ICE menu options, the user must click the left button while pointing to the "ICE" option on the "Main Menu". (See Figure A-23).



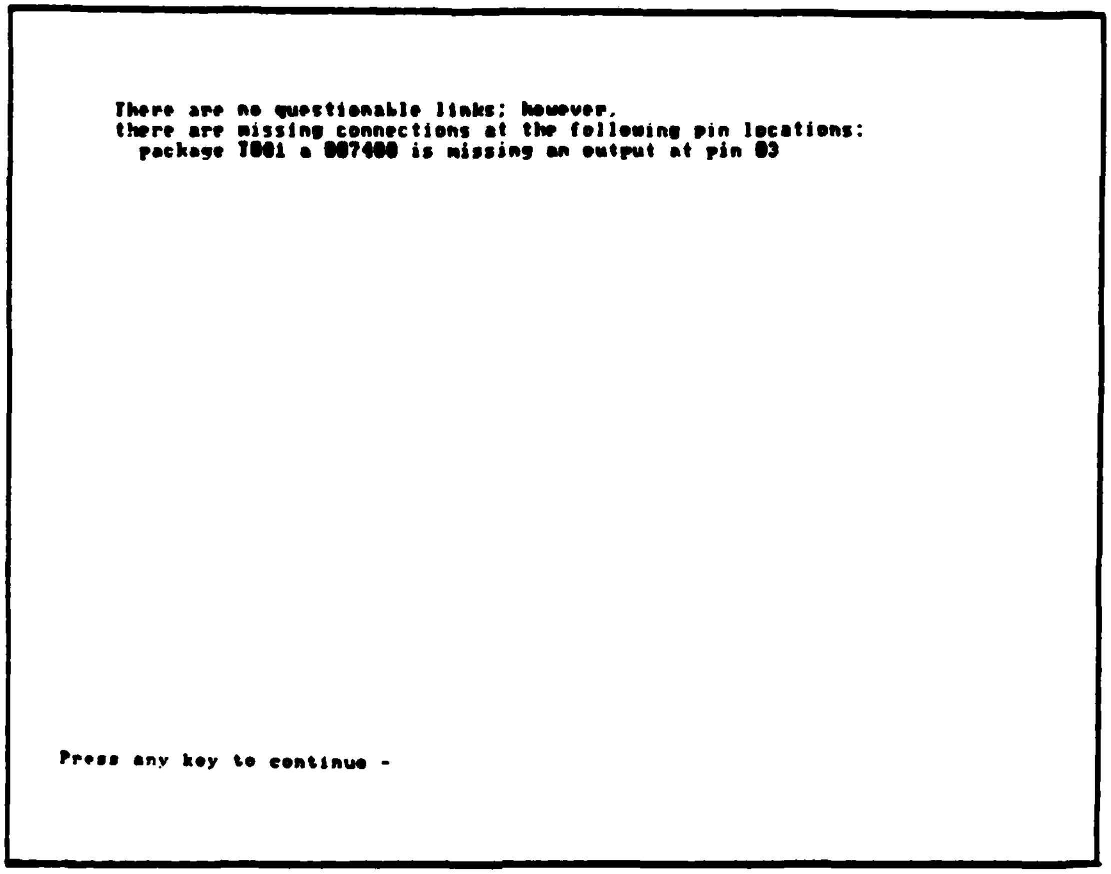
Sorry - need a circuit first - press any key to cont-

Figure A-23. Clicking on "ICE" option

If there is no circuit in progress at the time the user will be prompted with: "Sorry - need a circuit first - press any key to cont-" and the "ICE" title will be highlighted. Once a key is pressed the title will return to normal color. If a circuit is in progress the "ICE Menu" will replace the "Main Menu". (See Figure A-3).

Evaluating a circui*. To evaluate a circuit using the ICE program the user clicks left button while pointing to the "Execute" option in the "ICE Menu". The "Execute" title will remain highlighted until this option is terminated. The user will be advised: "Currently executing ICE - please standby". When ICE has finished, the user is prompted with: "ICE is finished - press any key to cont-". Once the user presses any key this option is terminated.

Viewing ICE results. To view the output results of ICE the user clicks the left button while pointing to the "View Rslt" option in the "ICE Menu". The current screen display will be replaced by a screen showing the results of the last ICE execution. (See Figure A-24). To return to the design screen, the user is prompted at the bottom of the screen (in yellow) to "press any key to cont-". If there exists more than one output screen display the user will need to press a key more than once, until the design screen reappears.



```
There are no questionable links; however,
there are missing connections at the following pin locations:
  package T001 a 007400 is missing an output at pin 03




Press any key to continue -
```

Figure A-24. ICE execution results

NOTE: THE RESULTS DISPLAYED ARE THE OUTPUTS OF THE LAST EXECUTION OF ICE. THESE RESULTS ARE NOT CHANGED UNTIL ICE IS EXECUTED AGAIN.

Interpreting ICE results. The information provided by ICE can be categorized into two categories: missing connections and questionable connections. If the ICE program finds any missing links, they are identified. If ICE finds any questionable connections, the end points of the link are specified. (See Figure A-24).

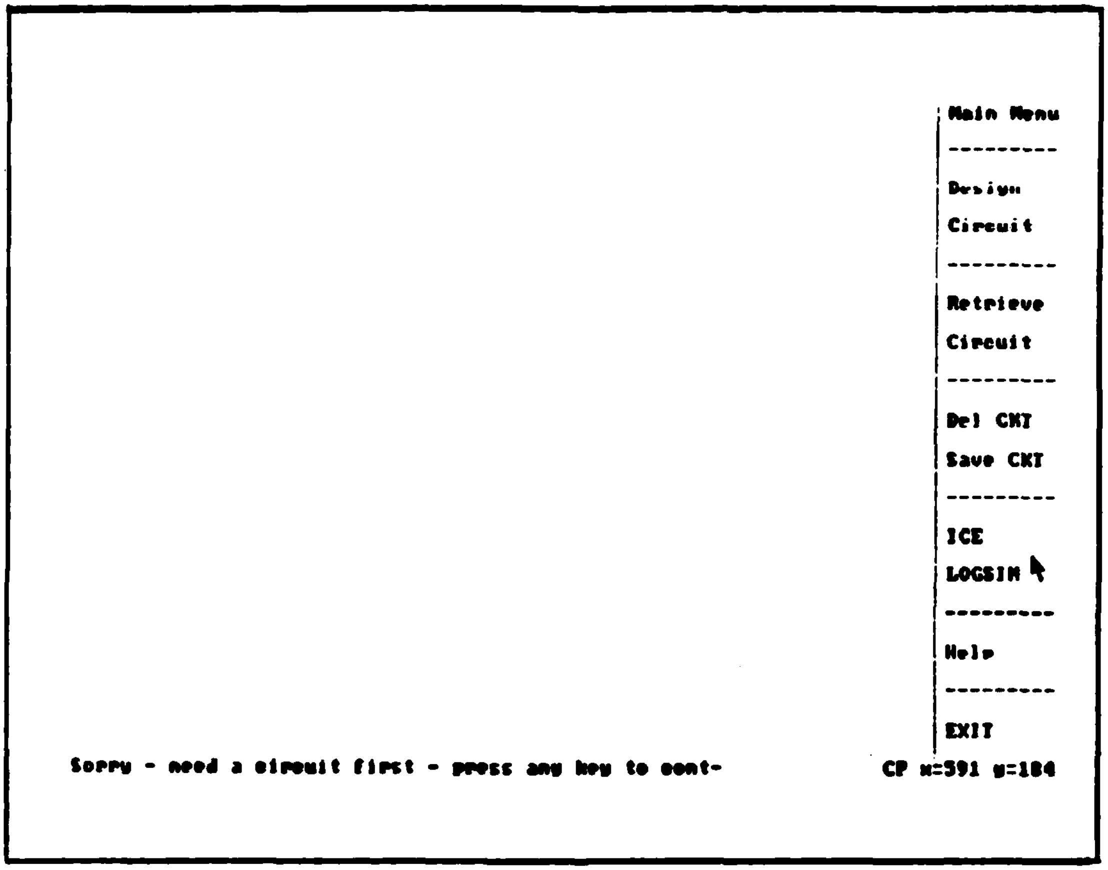
Printing ICE results. To get a printout of the ICE results described above, the user must click the left button on the

"Print" option of the "ICE Menu". The "Print" title will remain highlighted until this option is terminated. The user will be prompted with: "Ensure the printer is turned on and then press any key to cont-". Once the user presses any key, the following prompt is displayed: "Printing is being initiated - press RETURN to cont-". Once the information has been sent to the printer, the "Print" title will no longer be highlighted and the user may continue.

NOTE: THE USER MUST PRESS THE "RETURN" KEY TO ACTIVATE THE PRINTER DEFAULT OPTION, IF IT HAS NOT YET BEEN SET. IF IT HAS BEEN SET PREVIOUSLY, NO FURTHER ACTION IS REQUIRED. IF IT HAS NOT BEEN SET, THE USER WILL BE PROMPTED TO "PRESS ANY KEY TO CONTINUE".

VI. Using the LOGSIM tool:

To use any of the LOGSIM menu options, the user must click the left button while pointing to the "LOGSIM" option on the "Main Menu". (See Figure A-25).



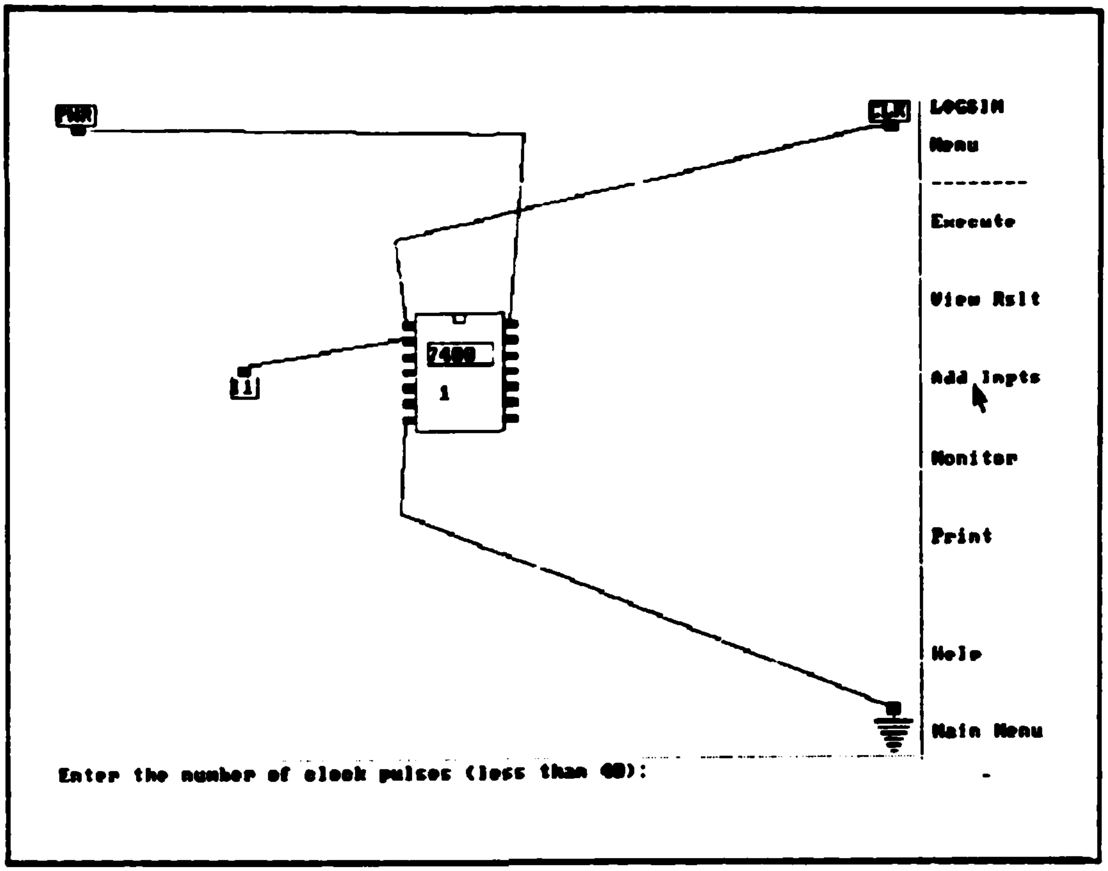
Sorry - need a circuit first - press any key to cont-

CP x=591 y=184

Figure A-25. Clicking on "LOGSIM" option

If there is no circuit in progress at the time the user will be prompted with: "Sorry - need a circuit first - press any key to cont-" and the "LOGSIM" title will be highlighted. Once a key is pressed the title will return to normal color. If a circuit is in progress the "LOGSIM Menu" will replace the "Main Menu". (See Figure A-4).

Adding input data stream(s). An input data stream must be specified for each established input port in the current circuit design. To specify the input stream the user must click on the "Add Inpts" option. Once the user does this, a prompt on the communications line will be provided. The user will be requested to specify the number of clock cycles (up to a maximum of 40) that he/she wishes the simulation to run. (See Figure A-26). If the user does not have any established input ports, the default value of 20 clock cycles is established. If there does exist any input ports, the title of the input port will be shown on the communications line and the user will need to enter the appropriate data stream of "1"s and "0"s. The number of bits in the data stream must be the same as the number of clock cycles or else the program will repeat the same input port title until the correct amount of inputs is provided. The user will need to specify the data stream for each input port once a valid number (i.e., between 1 and 40) of clock cycles has been specified.



Enter the number of clock pulses (less than 40):

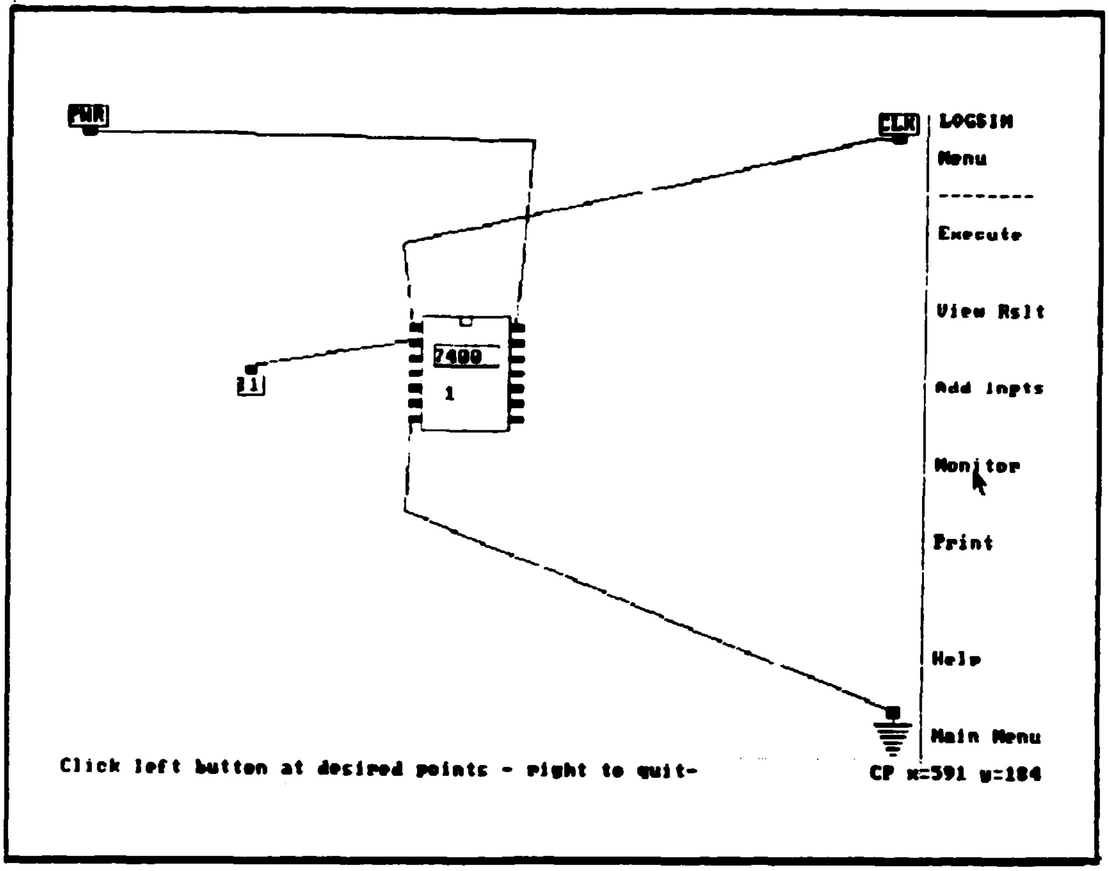
Figure A-26. Clicking on "Add Inpts" option

Error conditions:

If the user specifies an invalid number of clock cycles, the "Add Inpts" option is terminated.

If the user specifies an invalid number of bits for an input port (i.e., not the same number as the number of clock cycles) or an invalid input bit (i.e., not a "1" or "0") the program will provide an advisory identifying the discrepancy and prompt for a new input for the specific input port.

Identifying monitoring point(s). This option provides the user with capability to specify locations where output data is desired. The user must click on the "Monitor" option and the program prompts the user for the monitoring locations. (See Figure A-27). If the user clicks the right mouse button the "Monitor" option will be closed. All pins identified prior to closing will be used during the next execution of LOGSIM unless the "Monitor" option is selected again prior to execution. To specify a monitoring location, the user clicks on any pin. The cursor will momentarily disappear, while the pin location is tested for validity. If the cursor does not momentarily disappear, try clicking on the pin again.



Click left button at desired points - right to quit-

CP x=591 y=184

Figure A-27. Clicking on "Monitor" option

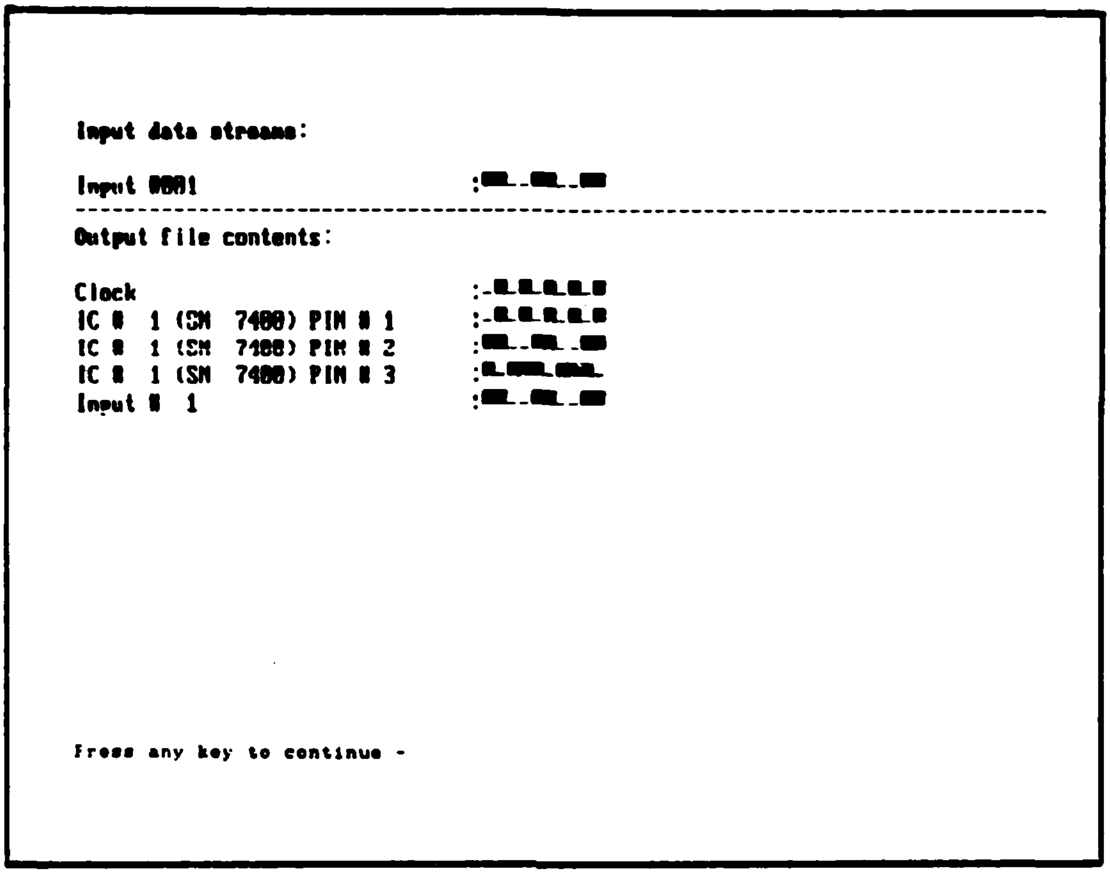
NOTE: THE USER CAN CLICK ON THE POWER AND GROUND PINS WITHOUT AN ADVISORY INDICATING THIS TO BE AN INVALID MONITORING POINT; HOWEVER, THESE POINTS WILL NOT SHOW UP IN THE RESULTS PROVIDED AFTER THE LOGSIM EXECUTION.

Error conditions:

If the user clicks on something other than a pin connection, the program will provide the advisory: "Not a valid monitoring location - press any key to cont-". Once the user presses any key the user can continue specifying monitor locations.

Simulating a circuit. To simulate a circuit using the LOGSIM program the user clicks the left button while pointing to the "Execute" option in the "LOGSIM Menu". The "Execute" title will remain highlighted until this option is terminated. The user will be advised: "Currently executing LOGSIM - please standby". When LOGSIM has finished, the user is prompted with: "LOGSIM is finished - press any key to cont-". Once the user presses any key this option is terminated.

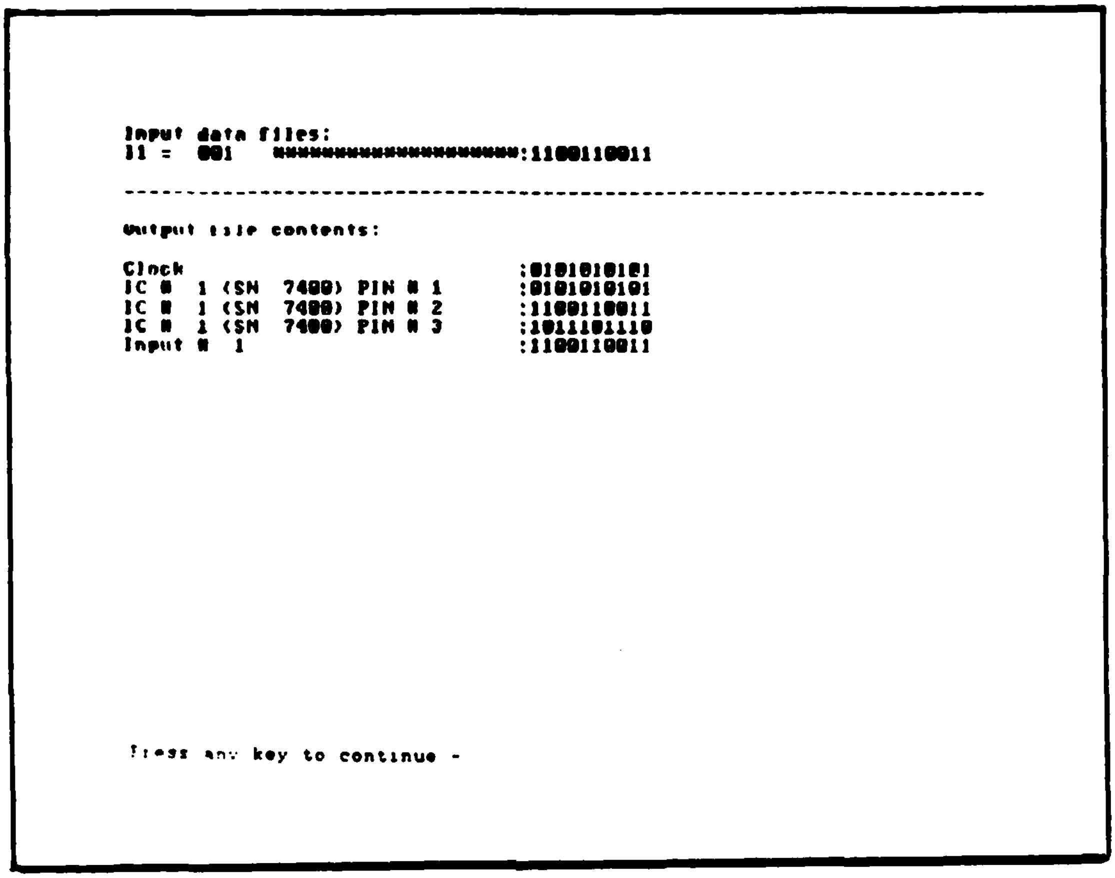
Viewing LOGSIM results. To view the output results of LOGSIM the user clicks the left button while pointing to the "View Rslt" option in the "LOGSIM Menu". The user will be prompted for the method of display (i.e., wave form or binary). If the user inputs a "W" or "w", the current screen display will be replaced by a screen showing the results of the last LOGSIM execution in wave form format. (See Figure A-28). To return to the design screen, the user is prompted at the bottom of the screen (in yellow) to "press any key to cont-". If there exists more than one output screen display the user will need to press a key more than once, until the design screen reappears.



```
Input data streams:

Input 0001                        :
-----------------------------------------------------------------------------
Output file contents:

Clock                             :
IC #  1 (SN  7400) PIN # 1        :
IC #  1 (SN  7400) PIN # 2        :
IC #  1 (SN  7400) PIN # 3        :
Input #  1                        :




Press any key to continue -
```

Figure A-28. LOGSIM execution results (wave form)

If the user inputs anything but a "W" or "w", the current screen display will be replaced by a screen showing the results of the last LOGSIM execution in binary format. (See Figure A-29). To return to the design screen, the user is prompted at the bottom of the screen (in yellow) to "press any key to cont-". If there exists more than one output screen display the user will need to press a key more than once, until the design screen reappears.



```
Input data files:
I1 =  001   ####################:1100110011

--------------------------------------------------------------------------------

Output file contents:

Clock                             :0101010101
IC #  1 (SN  7400) PIN # 1        :0101010101
IC #  1 (SN  7400) PIN # 2        :1100110011
IC #  1 (SN  7400) PIN # 3        :1011101110
Input #  1                        :1100110011
```

Press any key to continue -

Figure A-29. LOGSIM execution results (binary)

NOTE: THE RESULTS DISPLAYED ARE THE INPUT DATA STREAM(S) AND THE RESULTANT OUTPUTS AT THE SPECIFIED MONITORING POINTS AT THE TIME OF THE LAST EXECUTION OF LOGSIM. THE OUTPUT RESULTS ARE NOT CHANGED UNTIL LOGSIM IS EXECUTED AGAIN; HOWEVER, IF THE USER ADDS OR DELETES AN INPUT PORT, THE INPUT DATA STREAM(S) PORTION OF THE DISPLAY IS DELETED.

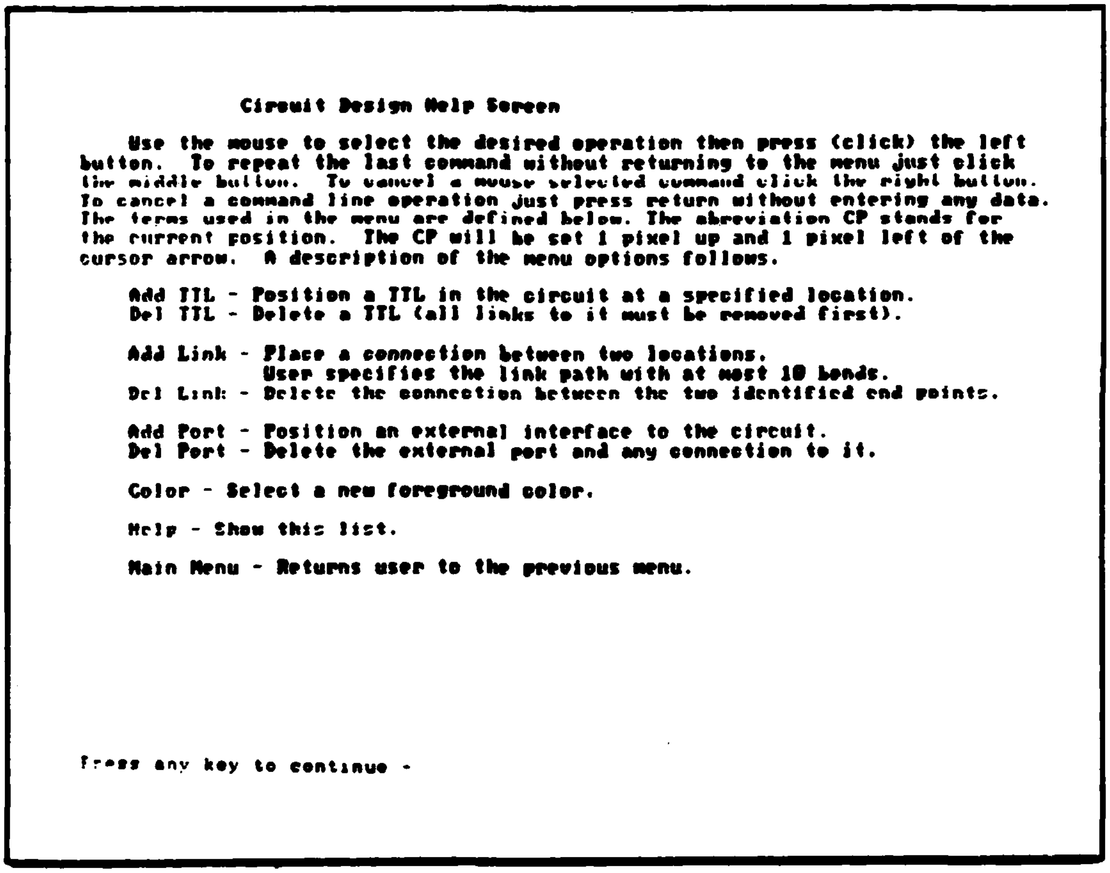
Interpreting LOGSIM results. The results of a LOGSIM execution contain the input data stream(s), if provided, and the output data stream(s) at each of the monitoring points. (See Figures A-28 and A-29).

Printing LOGSIM results. To get a printout of the LOGSIM results (in the binary format) described above, the user must click the left button on the "Print" option of the "LOGSIM Menu". The "Print" title will remain highlighted until this option is terminated. The user will be prompted with: "Ensure the printer is turned on and then press any key to cont-". Once the user presses any key, the following prompt is displayed: "Printing is being initiated - press RETURN to cont-". Once the information has been sent to the printer, the "Print" title will no longer be highlighted and the user may continue.

NOTE: THE USER MUST PRESS THE "RETURN" KEY TO ACTIVATE THE PRINTER DEFAULT OPTION, IF IT HAS NOT YET BEEN SET. IF IT HAS BEEN SET PREVIOUSLY, NO FURTHER ACTION IS REQUIRED. IF IT HAS NOT BEEN SET, THE USER WILL BE PROMPTED TO "PRESS ANY KEY TO CONTINUE".

VII. Getting help:

Each menu has a "Help" option. To get information applicable to the current menu options, the user clicks the mouse on the "Help" option and the current screen display will be replaced by a screen displaying the help information applicable to the current menu. To exit the "Help" screen and return to the circuit design window and current menu the user presses any key. (See Figure A-30).

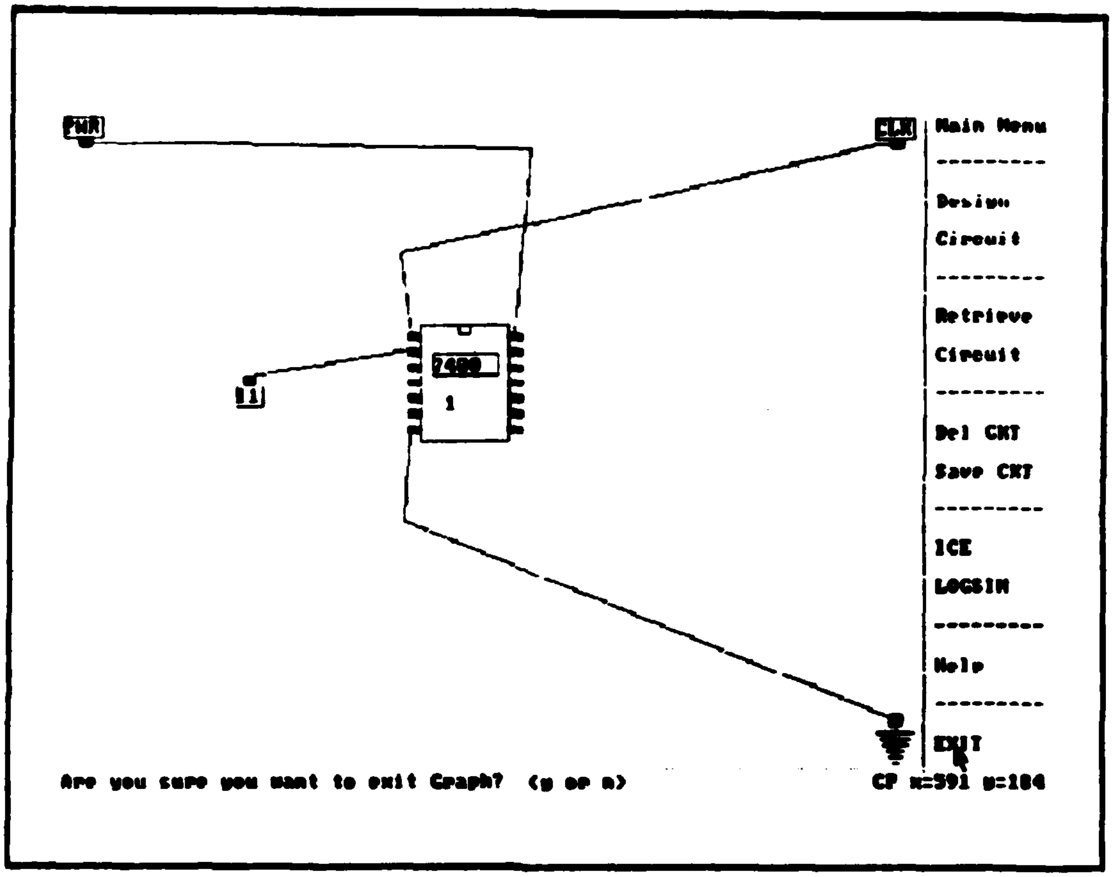


```
                Circuit Design Help Screen

    Use the mouse to select the desired operation then press (click) the left
button.  To repeat the last command without returning to the menu just click
the middle button.  To cancel a mouse selected command click the right button.
To cancel a command line operation just press return without entering any data.
The terms used in the menu are defined below. The abreviation CP stands for
the current position.  The CP will be set 1 pixel up and 1 pixel left of the
cursor arrow.  A description of the menu options follows.

    Add TTL - Position a TTL in the circuit at a specified location.
    Del TTL - Delete a TTL (all links to it must be removed first).

    Add Link - Place a connection between two locations.
               User specifies the link path with at most 10 bends.
    Del Link - Delete the connection between the two identified end points.

    Add Port - Position an external interface to the circuit.
    Del Port - Delete the external port and any connection to it.

    Color - Select a new foreground color.

    Help - Show this list.

    Main Menu - Returns user to the previous menu.




Press any key to continue -
```

Figure A-30. Example "Help" screen display

VIII. Exiting the graphics interface tool:

To exit the program the user clicks on the "Exit" option in the "Main Menu". The "Exit" option will be highlighted and the user will then be prompted with the following: "Are you sure you want to exit Graph? <y or n>". (See Figure A-31).



```
Are you sure you want to exit Graph?  (y or n)                CP x=591 y=184
```

Figure A-31. Clicking on "Exit" option

Entering a "Y" or a "y" results in an advisory on the communications line stating: "Removing temporary files - please be patient", the user is then returned to the host computer's operating system and the circuit design (if not saved prior to exiting) is lost.

If the user presses any other key the "Exit" option is ignored, "Exit" is no longer highlighted, the prompt is removed and the user is returned to the previous status.

IX. Design Considerations:

The following recommendations are presented to suggest methods of improving circuit designs:

1. Prior to placement of TTLs or input ports on the design window the user should consider the layout of his circuit to minimize connection overlap.

2. Match the color of the input port and all of its links. This will make it easier to trace wiring paths from the input port. Only the "Power", "Clock" and "Ground" icons have a fixed color (white). Colors must be selected prior to adding the TTL, input port, or link. Best method is to set the color then add the port and all the links from that port, then change to the next color and repeat for all remaining ports. This technique however requires the other end of the links (i.e., the TTL) to be already present, otherwise the links can not be connected.

3. Match the color of TTLs that have a related functionality. This may not always work since some TTLs can serve multiple functions simultaneously.

4. The user should save his circuit design regularly using the "Save CKT" option in the main menu. Using version numbers in the filename can be an effective tool for tracking design progress. For example, the following use of filenames for a binary to decimal converter: BCD1, BCD2, etc..

However, care should be taken in selecting filenames, the host computer's operating system has limitations on filename formats. For example, filenames in lower case may be converted to upper case.

5. The user should use the current position (CP) indicator in the lower right corner when placing TTLs and input ports on the screen because it will make designs neater. Same goes for drawing links, minor coordinate changes have a tendency to make the links look broken when in fact they are not.

Bibliography

1. Deloria, Capt Wayne C. A Digital Logic Simulator with Concurrent Programming Considerations. MS thesis, AFIT/GCS/ENG/87D-10. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1987.

2. Wagner, 1Lt Steven M. An Expert System for Discrete Component Digital Circuit Design. MS thesis, AFIT/GCS/ENG/87D-28. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1987.

3. Shaw, W.H. and George, B.L., "COMPUTER AIDED DESIGN IN GRADUATE ENGINEERING EDUCATION", Proceedings of the 1987 Engineering and Industrial Software Conference, May 1987.

4. Korth, Henry F. and Abraham Silberschatz. Database System Concepts. New York: McGraw-Hill Book Company, 1986.

5. Lorie, Raymond A. Issues in Database for Design Applications, 10 July 1981. IBM Research Laboratory, San Jose, California.

6. Ketabchi, Mohammad A. and Valdis Berzins. "Modeling and Managing CAD Databases," Computer: 93-102 (February 1987).

7. Woffinden, Duard S. Interactive Environment for a Computer-Aided Design System. MS Thesis, Naval Postgraduate School, Monterey, CA, 1984.

8. Ledgard, Henry. "Misconceptions in Human Factors," Abacus, 3,2: 21-27, 45 (Winter 1986)

9. Clay, Bruce. Air Force Institute of Technology/ENG Graphics Package, Wright-Patterson AFB OH, 4 May 1987.

10. Pressman, Roger S. Software Engineering: A Practioner's Approach (Second Edition). New York: McGraw-Hill Book Company, 1987.

VITA

Captain Charles A. Adams Jr. was born on 15 September 1955 in Fuerstenfeldbruck, Germany. He graduated from California State University at Sacramento in 1978, with a Bachelor of Arts Degree in Mathematics. He received his commission through the Officer Training School (OTS) in May 1978. After graduation from OTS, he was stationed at Keesler Air Force Base (AFB), Mississippi, where he attended the Communications Electronics Officer School. Upon graduation in January 1979, he was assigned as Chief, Combat Crew Communications Branch, 2034th Communications Squadron (AFCC), Mather AFB, California. In March 1982, he was assigned to the 1st Aerospace Communications Group (AFCC), Offutt AFB, Nebraska, where he became Chief, Airborne Command Post Standardization/Evaluation Communications Operations, supporting Strategic Air Command. In June 1984, he was assigned as Communications Electronics Inspector, Office of the Inspector General, Headquarters Strategic Air Command (SAC), Offutt AFB, until entering the School of Engineering, Air Force Institute of Technology, in May 1986.

Permanent address: 3850 Berrendo Dr.
Sacramento,
California, 95864

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS
A188831

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release;
distribution unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
AFIT/GCS/ENG/87D-1

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
School of Engineering

6b. OFFICE SYMBOL *(If applicable)*
AFIT/ENG

7a. NAME OF MONITORING ORGANIZATION

6c. ADDRESS *(City, State, and ZIP Code)*
Air Force Institute of Technology
Wright-Patterson AFB OH 45433-6583

7b. ADDRESS *(City, State, and ZIP Code)*

8a. NAME OF FUNDING / SPONSORING ORGANIZATION

8b. OFFICE SYMBOL *(If applicable)*

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS *(City, State, and ZIP Code)*

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.

11. TITLE *(Include Security Classification)*
A DIGITAL CIRCUIT DESIGN ENVIRONMENT (UNCLASSIFIED)

PERSONAL AUTHOR(S)
Charles A. Adams, Jr., B.A., Capt, USAF

13a. TYPE OF REPORT
MS Thesis

13b. TIME COVERED
FROM ________ TO ______

14. DATE OF REPORT *(Year, Month, Day)*
1987 December

15. PAGE COUNT
146

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP
12	05	

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Computer Graphics, Computer Aided Design, Graphics

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

Thesis Chairman: Bruce L. George, Captain, USAF
Assistant Professor of Electrical Engineering

Approved for public release: IAW AFR 190-1.
31 Dec 87
LYNN E. WOLAVER
Dean for Research and Professional Development
Air Force Institute of Technology (~~AGC~~)
Wright-Patterson AFB OH 45433

. DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED □ SAME AS RPT. □ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
Bruce L. George,Captain, USAF

22b. TELEPHONE *(Include Area Code)*
(513) 255-2024

22c. OFFICE SYMBOL
AFIT/ENG

Previous editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

19 (cont)

This thesis effort outlines the design and implementation of a graphics oriented digital design environment. This graphics environment provides the user with the tools to design digital circuits and then interface the user's circuit design with other tools (i.e., the Logic Simulator (LOGSIM) and Interconnect Expert (ICE) programs). This research paper presents the reasoning for the development of such a tool. The development of this tool involves reviewing database designs, specifically those oriented toward Computer Aided Design (CAD), and human-computer interface considerations. The paper also presents a recap of the features necessary for the graphics oriented environment. The detailed design of the program is presented along with the description of the file structures, which provide the underlying database. The limitations encountered and the results of the testing are addressed. In addition, a comprehensive user's manual is included for the operation of the graphics oriented interface. Finally, a listing of recommended follow-on efforts are presented with the conclusions of this thesis effort.

END
DATE
FILMD
3-88
DTIC